PARALLELS BETWEEN GLUCONEOGENESIS AND SYNCHRONOUS MACHINES

Monendra Grover

Amity Institute of Biotechnology, Amity University, NOIDA
Current Address : National Bureau of Plant Genetic Resources, Pusa Campus, New Delhi

ABSTRACT

Biological diversity particularly at molecular level is astounding and can be used for rational manipulation of biological organisms. To analyze molecular diversity in its full scope computational models of biological organisms and biochemical pathways are indispensable. Engineering Sciences can be of great help in construction of these models. The central feature of modern engineering has been system level design. The differences between biological systems and engineering systems are notable, particularly at the molecular and device level. However, convergent evolution is thought to yield remarkable similarities at higher levels of organization. Here we compare an electrical engineering system with a biological system. We take the example of synchronous machines and gluconeogenesis. A biochemical system like gluconeogenesis is not just an assembly of enzymes. In addition to the list which catalogs the individual components, it is essential to understand how individual components dynamically interact during such operation. In such an attempt the concept of computational complexity is applied to both gluconeogenesis and synchronous machines The Church-Turing hypothesis is used as the basis to construct models of gluconeogenesis and synchronous machines . It is shown that both synchronous machines and gluconeogenesis accept context sensitive languages and models of computation of both the systems are universal. Thus we conclude that for construction of computational models of biochemical diversity in biological organisms, engineering systems can provide important clues.

Keywords : System, synchronous machines, gluconeogenesis

INTRODUCTION

Ludwig von Bertalanffy used the word systems in 1967 in introduction to his book General System Theory. [1] According to Bertalanffy obsolete science "tried to explain observable phenomenon by reducing them to an interplay of elementary units investigatable independently of each other". Modern science on the other hand recognized the relevance of "wholeness" defined as "problems of organization, phenomena not resolvable into local events, dynamic interactions manifest in the difference of behavior of parts when isolated or in higher configuration etc; in short 'systems' of various orders not understandable by investigation of their respective parts in isolation".

In biological sciences system level understanding has been a recurrent theme since the days of Norbert Weiner[2], [3]. System level approaches in biology are now receiving renewed attention. While an understanding of constituent enzymes in a biochemical process is important the focus should also be on understanding a system's structure and dynamics. Because a biochemical system is just not an assembly of enzymes, its properties cannot be fully understood merely by drawing diagrams of their interconnections. Identifying all enzymes in a biochemical/physiological processes is like listing all the parts of a sophisticated machine. While such a list provides a catalog of the individual components, by itself it is not sufficient to understand the complexity underlying the system.[4]. Such a description provides limited knowledge of how changes to one part of a system may affect other parts. To understand how a particular system functions the question that how the individual components dynamically interact during operation must be addressed.

System level design has also consistently been at the core of modern engineering, motivating most sophisticated theories in controls, information and computation. The differences between biological and engineering systems are conspicuous, particularly at the molecular and device level. Nevertheless convergent evolution, a concept widely used in both engineering and evolutionary biology is thought to yield remarkable similarities at higher levels of organizations. As a result of advanced controls and embedded networking all technologies are evolving similarly. Csete and Doyle (2002) have claimed that technological evolution of complexity is convergent with that of biology. Here we compare a biological system (gluconeogenesis) with that of an electrical engineering system (synchronous machine) with reference to their computational complexity.

RESULTS AND CONCLUSIONS

SYNCHRONOUS MACHINES AND GLUCONEOGENESIS BEHAVE AS AUTOMATA WHICH ACCEPT FORMAL LANGUAGES

The theory of formal languages is an area with a number of applications in computer science. In the early 1950s linguists were trying to define precisely valid sentences and give structural descriptions of sentences. They wanted to define a formal grammar (i.e. to describe the rules of grammar in a rigorous mathematical way) to describe English. It was thought that such a description of natural languages would make language translation using computers easy. It was Noam Chomsky who gave a mathematical model of a grammar in 1956. Although it was not useful for describing natural languages such as English, it turned out to be useful for computer languages. In fact the Backus-Naur form used to describe ALGOL followed the definition of grammar (a context free grammar) given by Chomsky.

Using this theory and based on the chemical equations describing glconeogenesis we can express gluconeogenesis as a context sensitive language acceptance problem. The chemical reactions can be envisaged as performing a transformation on words of a context sensitive language (transformation on the reactants to produce products). We propose that the process of gluconeogenesis behaves like an automaton that computes (or accepts) a formal language. The gluconeogenesis can be represented as:

$$2\ \text{Pyruvate} + 4\ \text{ATP} + 2\text{GTP} + 2\text{NADH} + 4H_2O \longrightarrow \text{glucose} + 4\text{ADP} + 2\text{GDP} + 6P_i + 2\ NAD^+ + 2H^+$$

This can be represented as transformation on words:

$$(\text{pyruvate})^{k0}\ (\text{ATP})^{k1}\ (\text{GTP})^{k2}\ (\text{NADH})^{k3}\ (H_2O)^{k4} \longrightarrow (\text{pyruvate})^{k0-2n}\ (\text{ATP})^{k1-4n}\ (\text{GTP})^{k2-2n}\ (\text{NADH})^{k3-2n}\ (H_2O)^{k4-4n} (\text{glucose})^{n}\ (\text{ADP})^{4n}\ (\text{GDP})^{2n}\ (P_i)^{6n}\ (NAD^+)^{2n}\ (H^+)^{2n}$$

Notation x^y denotes y molecules (copies) of x. Also $k_0, k_1, k_2, k_3, k_4 \varepsilon N$ are the number of units of pyruvate, ATP, GTP, NADH, H_2O molecules respectively available to system during transformation. Small case $n \varepsilon N$ is the number of glucose molecules produced during transformation.

Similar formalism has been used to describe process of biological photosynthesis by [5]. However our formalism is used for gluconeogenesis and synchronous machines and the definition of components of our model are conceptually different from those used by above-mentioned authors.

In a similar vein the synchronous machines can be envisaged as accepting a language and performing a transformation on words of context sensitive language (the transformation of a three phase short circuit into its output).

It has been mentioned before that both synchronous machines and gluconeogenesis behave as automata. The most general definition of an automaton is a system where energy, materials and information are transformed, transmitted and used for performing some functions without direct participation of man. In computer science the term automaton means discrete automaton and is defined in a more abstract way . The main characteristics of automaton are

Input: At each of the discrete instants of time $t_1, t_2, \ldots t_m$ the Input values $I_1, I_2, \ldots I_p$, each of which can take a finite number of fixed values from the input alphabet Σ, are applied to the input side .
Output $O_1, O_2, \ldots, O_q$ are the outputs of the model, each of which can take a finite number of fixed values from an output O.
States: At any instant of time the automaton can be in one of the states q_1, q_2, q_n
State relation: The next state of an automaton at any instant of time is determined by the present state and the present input.
Output relation: The output is related to either state only or to both the input and the state.
We define these parameters for gluconeogenesis and synchronous machine later in the paper.

COMPUTATION IN GLUCONEOGENESIS AND SYNCHRONOUS MACHINES

In the early 1930s mathematicians were trying to define effective computation. Turing in 1936, Church in 1933, Kleene in 1935 and Schonfinkel in 1965 gave various models using the concept of Turing machines, lambda calculus, combinatory logic, post-systems and μ -recursive functions respectively. It is relevant to note that these were formulated much before the electronic computers were devised. Although these formalisms, describing effective computations are different they turn to be equivalent. Among these formalisms, the

Turing's formulation is accepted as a model of algorithm or computation. The Church Turing thesis states that any algorithmic procedure that can be carried out by human beings/computer, can be carried out by a Turing machine. It has been universally accepted by computer scientists that the Turing machine provides an ideal theoretical model of a computer.

I propose that the Gluconeogenesis and synchronous machines accept a language that is at least as hard as the textbook [6] context sensitive language and is itself context sensitive. Thus these problems require a model of computation that has at least the power of a linear bounded automaton . (A Turing machine in which a linear function is used to restrict ('bound') the length of the tape. As mentioned before similar formalism has been used for photosynthesis. [5]

Work in chemical computation has followed four different paths. It is one of the natural extensions of work on information and thermodynamics, which date back to Maxwell demon arguments. It is also a corollary of the application of dynamical systems theory to chemical reactions, in particular logic networks stemming from bistable reaction systems. A lot of work has been devoted to trying to devise nonstandard computational architectures, and finally the enzymatic cascades of signaling have been discussed with reference to computation. [7]

In this paper we argue that the models of computation of generalized gluconeogenesis and synchronous machines are universal. It has been proved by Minsky that even a single automaton with no more than two counters is universal [8]. This means that respiration and synchronous machines in their general form are capable of computing any function that is computable by a digital electronic computer or Turing machine. We propose that this generalized model of respiration and synchronous machines is a reasonable generalization of a machine to accept the language as described earlier. Similar conclusions have been derived for photosynthesis [5].

GLUCONEOGENESIS AND SYNCHRONOUS MACHINES AS LINEAR BOUNDED AUTOMATA

As mentioned above we propose that the Gluconeogenesis and synchronous machines accept a language that is at least as hard as the textbook [6] context sensitive language and is itself context sensitive. We have also mentioned that these problems, therefore, require a model of computation that has at least the power of a linear bounded automaton .

The model of Linear Bounded Automaton

This model is important because (a) the set of context-sensitive languages is accepted by the model and (b) the infinite storage is restricted in size but not in accessibility to the storage, in comparison with the Turing machine model. It is called the linear bounded automaton (LBA) because a linear function is used to restrict (to bound) the length of the tape.

A linear bounded automaton is a nondeterministic Turing machine, which has a single tape whose length is not infinite but bounded by a linear function of the length of the input string. The models can be described formally by the following set format:

$M= (Q, \Sigma, \Gamma, \delta, q_0, b, , \$, F)$

1. Q is a finite nonempty set of states
2. Γ is a finite nonempty set of tape symbols
3. $b \in \Gamma$ is the blank
4. Σ is a nonempty set of input symbols and is a subset of Γ and $b \notin \Sigma$
5. δ is the transition function mapping (q,x) onto (q', y, D) where D denotes the direction of movement of R/W head; D=L or R according as the movement is to the left or right.
6. $q_0 \in Q$ is the initial state
7. $F \subseteq Q$ is the set of final states.

All the symbols have the same meaning as in the basic model of Turing machines with the difference that the input alphabet Σ contains two special symbols and $.

Synchronous machines as linear bounded automata

The inputs of the synchronous machines (Fig.1), [9] can be envisaged as being inscribed on a linear tape which is converted into output (Fig. 2 and 3), [9] by the finite control (components of synchronous machines described earlier). Since the input tape is not infinite, synchronous machines can be compared to linear bounded automata. Now we define various parameters for synchronous machines

Q is the set containing resting state of the synchronous machine (no input) and the state of synchronous machine when three-phase fault is applied

Σ is the short circuit currents described in Fig.1

δ is the set containing equations describing synchronous machines.
$q_0 \in Q$ represents the state of synchronous machine when no input is given
F is the final state of synchronous machines (same as the initial state q_0)
Δ is the output of synchronous machines in response to a three phase short circuit shown in Fig.2 and 3
λ is a mapping from Q to Δ

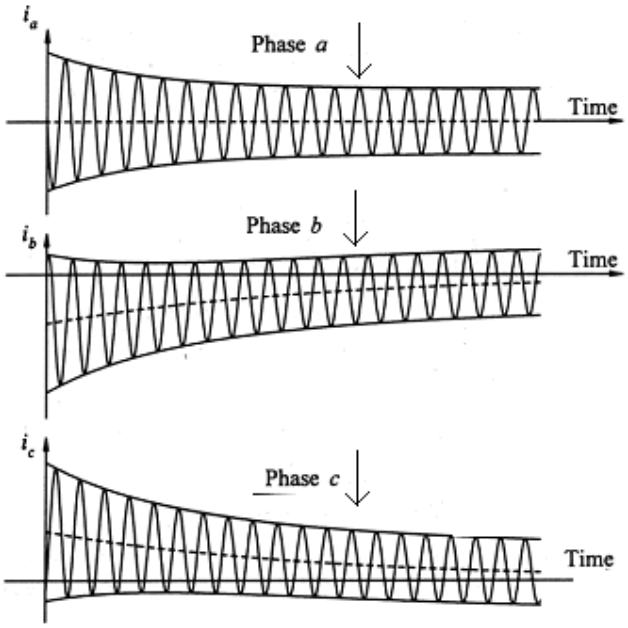

Figure 1 : Short circuit current in three phases

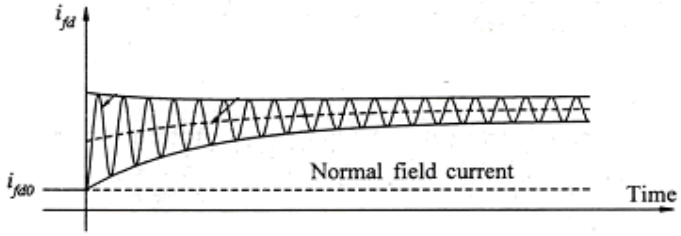

Fig.2 Field current response following stator short-circuit

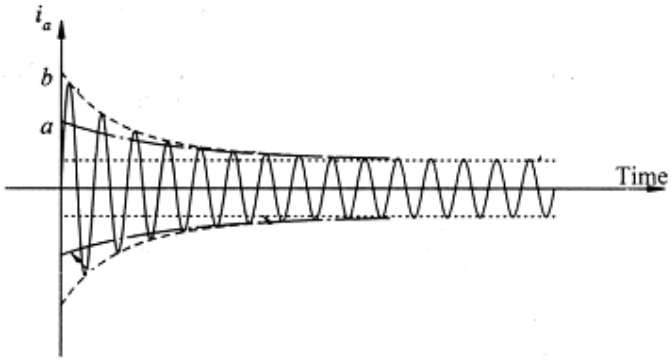

Fig.3 Fundamental frequency component of armature current

Gluconeogenesis as linear bounded automata

We define various parameters for gluconeogenesis:

The various symbols are as described above. Also
Δ is the output alphabet and
λ is a mapping from Q X Σ to Δ giving the output associated with each state. That is λ (q,a) gives the output associated with the transition from state q on input a. the output of M in response to input a_1, a_2, … a_n is λ (q_0, a_1), λ (q_1,a_2).. λ (q_{n-1}, a_n) where q_0, q_1…q_n is the sequence of states such that δ (q_{i-1}, a_i)= q_i for $1 \leq i \leq n$.

Gluconeogenesis can be divided into two automata: (a) the reactions occurring inside mitochondria and (b) the reactions occurring in the cytoplasm.
For automaton A the representation would be

Q = Various combinations of 'Resting' and 'active' state of enzymes involved in gluconeogenesis (1) pyruvate decarboxylase (2) malate dehydrogenase
The various states of the machine can be
Q_0: 1,2
Q_1: 1*, 2
Q_2: 1,2*
Q_3: 1,2

Asterisked numerals indicate enzymes 'in action'. Rest of the numerals indicate resting state of the enzymes.
Σ = {Pyruvate, Oxaloacetate }
δ = transition function which maps substrates of enzymatic reactions to their products
Δ is the (Oxaloacetate, Malate)
$q_0 \in$ Q defined above
λ is mapping from Q X Σ to Δ

The output of the machine in response to (Pyruvate, Oxaloacetate) is λ (q_1 pyruvate), λ (q_2, oxaloacetate),
λ (q_1 pyruvate) = oxaloacetate
λ (q_2, oxaloacetate) = malate
Also
δ (q_1 pyruvate) = q_2
δ (q2, oxaloacetate) = q3

For automaton B
Automaton B would be represented as
Q = Various combinations of 'Resting' and 'active' state of enzymes involved in automata B. 2: Malate dehydrogenase, 3: Phosphoenolpyruvate carbxykinase, 4: Enolase, 5: Phosphoglyceromutase, 6: Phosphoglycerate kinase, 7: Glyceraldehyde 3-Phosphate dehydrogenase 8: Aldolase 9: Fructose 1,6 bisphosphatase, 10: Phosphoglucose isomerase, 11: Glucose-6-phosphatase
Q_0 =2,3,4,5,6,7,8,9,10,11
Q_1 = 2*, 3,4,5,6,7,8,9,10,11
Q_2 = 2,3*, 4,5,6,7,8,9,10,11
Q_3 = 2,3,4*, 5,6,7,8,9,10,11
Q_4 = 2,3,4,5*, 6,7,8,9,10,11
Q_5 = 2,3,4,5,6*, 7,8,9,10,11
Q_6 = 2,3,4,5,6,7*, 8,9,10,11
Q_7 = 2,3,4,5,6,7,8*, 9,10,11
Q_8 = 2,3,4,5,6,7,8,9*, 10,11
Q_9 = 2,3,4,5,6,7,8,9,10*, 11
Q_{10}= 2,3,4,5,6,7,8,9,10,11*
Q_{11}= 2,3,4,5,6,7,8,9,10,11

Σ = {malate, oxaloacetate, phosphoenolpyruvate, 2-phosphoglycerate, 3-phosphoglycerate, 1,3 bisphosphoglycerate, glyceraldegyde-3-phosphate, Dihydroxyacetone phosphate, fructose-1,6 bisphosphate, Fructose-6-phosphate, glucose-6-phosphate}
δ = transition function which maps substrates of enzymatic reactions to their products (See above)
Δ is the {oxaloacetate, phosphoenolpyruvate, 2-phosphoglycerate, 3-phosphoglycerate, 1,3 bisphosphoglycerate, Glyceraldehyde-3-phosphate, Fructose-1,6 bisphosphate, Fructose-6-phosphate, Glucose-6-phosphate, Glucose}
λ is a mapping from Q X Σ to Δ

The output of the machine in response to {malate, oxaloacetate, phosphoenolpyruvate, 2-phosphoglycerate, 3-phosphoglycerate, 1,3 bisphosphoglycerate, glyceraldegyde-3-phosphate, Dihydroxyacetone phosphate, fructose-1,6 bisphosphate, Fructose-6-phosphate, g;ucose-6-phosph} can also be represented as λ (q_1, malate), λ (q_2, oxaloacetate), λ (q_3, phosphoenolpyruvate), λ (q_4, , 2-phosphoglycerate), λ (q_5, 3-phosphoglycerate), λ (q_6, 1,3 bisphosphoglycerate), λ (q_7, glyceraldehydes-3-phosphate), λ (q_8, fructose 1,6 bisphosphate), λ (q_9, fructose-6-phosphate), λ (q_{10}, glucose-6-phosphate)

λ (q_1, malate) = oxaloacetate
λ (q_2, oxaloacetate) = phosphoenolpyruvate
λ (q_3, phosphoenolpyruvate) = 2-phosphoglycerate
λ (q_4, , 2-phosphoglycerate) = 3-phosphoglycerate
λ (q_5, 3-phosphoglycerate) = 1,3 bisphosphoglycerate
λ (q_6, 1,3 bisphosphoglycerate) = glyceraldehyde-3-phosphate
λ (q_7, glyceraldehyde-3-phosphate) = fructose-1,6 bisphosphate
λ (q_8, fructose-1,6 bisphosphate) = fructose-6-phosphate
λ (q_9, fructose-6-phosphate) = glucose-6-phosphate
λ (q_{10}, glucose-6-phosphate)= Glucose

Also
δ (q_1, malate) = q_2
δ (q_2, oxaloacetate) = q_3
δ (q_3, phosphoenolpyruvate) = q_4
δ (q_4, , 2-phosphoglycerate) = q_5
δ (q_5, 3-phosphoglycerate) = q_6
δ (q_6, 1,3 bisphosphoglycerate) = q_7
δ (q_7, glyceraldehyde-3-phosphate) = q_8
δ (q_8, fructose-1,6 bisphosphate) = q_9
δ (q_9, fructose-6- phosphate) = q_{10}
δ (q_{10}, glucose-6-phosphate) = q_{11}

Our formalism can be used to show that context sensitive process of Gluconeogenesis and response of synchronous machines is composed of several simpler processes that in turn accept context sensitive languages. Gluconeogenesis as mentioned above can be defined into two automata A and B. The synchronous machines can be divided into stator, rotator, armature winding and field winding components. The combination and communication of these processes results in greater computational power of the system. An important role is played by modularity in biological as well as engineering systems. Modules generally are defined as components, parts or subsystems of a larger system that contain some or all of the following features: (i) identifiable interfaces (usually involving protocols), (ii) can be modified and evolved somewhat independently (iii) facilitate simplified or abstract modeling (iv) maintain some identity when isolated or rearranged, yet (v) derive additional identity from the system. [10]. Universal principles are suggested by the organization and design of technologies (such as synchronous machines), which are relevant to biology. These principles link modularity with the robust yet fragile nature of complex systems.

Protocols are also important in both biological and engineering systems. If modules are ingredients, parts, components, subsystems and players, then protocols describe the corresponding recipes, architectures, rules and interfaces. Protocols are rules that prescribe allowed interfaces between modules, permitting system functions that could not be achieved by isolated modules. Protocols also facilitate the addition of new protocols and organization into collections of mutually supportive protocols [10]. In our case protocols involving communication between automaton A and automaton B in gluconeogenesis and between stator, rotor, armature winding and field winding components serve an important function.

REFERENCES

[1] Von Bertalanffy, L. (1969) General system Theory, Foundations, Development, Applications (George Braziller, New York)
[2] Weiner, N. (1948) Cybernetics or Control and Communication in the Animal and the machine (MIT Press, Cambridge, MA, 1948)
[3] Savageau, M.A. , Biochemcial systems theory (Addison-Wesley, Reading, MA, 1976)
[4] Kitano, H. (2002) Systems biology : a brief overview. Science 295: 1662-1664
[5] Murphy, N., Woods, D. and Naughton , T.J. (2005) On the computational power of photosynthesis. Technical Report Series NUIM-CS-TR-2005-03 Department of Computer Science, National University of Ireland, Maynooth
[6] Sipser, M. (1996) Introduction to theory of computation. PWS publishing
[7] Magnasco, O.M. (1997) Chemical kinetics is Turing universal, Phy. Rev. Lett. 78(6): 1190-1193

[8] Minsky, M.L. (1967) Computation, finite and infinite machines. Prentice Hall.
[9] Kundur, P. (2006) Power system stability and control. Tata McGraw-Hill
[10] Csete, M.E. and Doyle, J.C. (2002) Reverse engineering of biological complexity. Science 295:1664-16