REVIEW

Intelligently Deciphering Unintelligible Designs: Algorithmic Algebraic Model Checking in Systems Biology

BUD MISHRA[1,2*]

[1] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA 10012.

[2] NYU School of Medicine, 550 First Avenue, New York, NY, USA 10016.

May 16, 2008;– Last modified by Bud

ABSTRACT
Systems biology, as a subject, has captured the imagination of both biologists and systems scientists, alike. But what is it? This review provides one researcher's somewhat idiosyncratic view of the subject, but also aims to persuade young scientists to examine the possible evolution of this subject in a rich historical context. In particular, one may wish to read this review to envision a subject built out of a consilience of many interesting concepts from systems sciences, logic and model theory, and algebra, culminating in novel tools, techniques and theories that can reveal deep principles in biology—seen beyond mere observations. A particular focus in this review is on approaches embedded in an embryonic program, dubbed "Algorithmic Algebraic Model Checking," and its powers and limitations.

1 HYPOTHESES NON FINGO: HOOKE AND NEWTON

Over the last few years, Sir Robert Hooke, a somewhat maligned, but still a very fascinating English experimental scientist and the first secretary of the Royal Society, had begun to feature unexpectedly prominently in practically all my public presentations on Systems Biology. Initially, what had attracted me to the story of Hooke, was the uncanny resemblance he bore to some of our present-day scientists in terms of their insistence on data, observations and hypotheses, their apparent non-rigorous and intuitive approaches to scientific questions, but most inexplicably, their protracted and debilitating open rivalries over the questions of recognition. But, as I learned more about Hooke's life and views, it also became clearer that his indirect influence on the way we think about science today is only surpassed by the opinions of only a handful of other contemporary thinkers, with some of whom Hooke fought bitter and hopeless semi-philosophical battles. They have, thus, unwittingly lent us a useful perspective that is worth examining with some care. How the emerging field of systems biology could establish itself, how it should face its trials and tribulations along the way, and how it could be a significant component of the "new new" biology, etc., could all be examined from the points of view of these 17th century scientists—a perspective that remains anachronically and peculiarly relevant even today.

Robert Hooke (1635-1703) was an experimental scientist, mathematician, architect, and astronomer. He was also the first Secretary of the Royal Society from 1677 to 1682, and because of his wide ranging interests, Hooke has been variously described as the "England's Da Vinci." His work Micrographia of 1665 contained his microscopical investigations, which included the first identification of biological cells, an enduring discovery that has maintained its centrality in subsequent developments in biology for more than three centuries. In his drafts of Book II, Newton had referred to him as the most illustrious Hooke—"Cl[arissimus] Hookius." However, not long after, Hooke became involved in a bitter dispute with Sir Isaac Newton over the priority of the discovery of the inverse square law of gravitation. In a letter Hooke wrote to Halley, he complained about omission of credit given to his discovery of the properties of gravity, "which of late Mr. Newton has done me the favour to print and publish as his own inventions." In response Newton wrote back to Halley, "Now is this not very fine? Mathematicians that find out, settle & do all the business

[*] To whom correspondence should be addressed. E-mail: mishra@nyu.edu

must content themselves with being nothing but dry calculators & drudges I beleive[sic] you would think him a man of a strange unsociable temper"—perhaps still a common protest of many disgruntled mathematicians whose contributions to science have failed to be adequately recognized. In a more well-known letter that Newton wrote directly to Hooke, he famously said, "If I have seen further[sic] than other men, it is because I have stood on the shoulders of giants"—where, of course, the giants Newton was alluding to were Kepler and Galileo, and not the dwarfish, small-minded and short-tempered likes of Hooke! When Christopher Wren was brought in to resolve this rather strangely English war-of-words, Wren diplomatically described the disagreement using Clairaut's characterization of "the great distance between a glimpsed truth and a demonstrated truth"—raising perhaps, the question of relative roles that should be ascribed to the inductive hypothesis-driven science in relation to the deductive principle-driven science.

What is the nature of "TRUTH" in biology, and how is it to be sought? Hooke saw biology as an observational science; he wrote in Micrographia, "The truth is, the science of Nature has already been too long made only a work of the brain and the fancy. It is now high time that it should return to the plainness and soundness of observations on material and obvious things," —a view supporting hypothesis-driven experimentation that advances science through steps of falsification or validation. Newton, on the other hand, championed a search for deep and unifying principles. Newton shunned hypotheses; his motto stated in Principia was "Hypotheses non fingo." ("I feign no hypotheses.") Newton's viewpoints are probably best stated by his most ardent disciple, Halley; in his rather ornately titled essay 'The true Theory of the Tides, extracted from that admired Treatise of Mr. Isaac Newton, Intituled, Philosophiae Naturalis Principia Mathematica,' he wrote the following: "Truth being uniform and always the same, it is admirable to observe how easily we are enabled to make out very abstruse and difficult matters, when once true and genuine Principles are obtained."

Biology still remains an observational science; it continues to move through the toils of a vast army of scientists each examining a small subsystem of a favored organism, as the scientists sharpen their intuitions, build upon guesses, conjectures, and hypotheses, and refine their ideas in many small steps—occasionally interrupted by a great leap, a grand vision or a comprehensive shift in paradigm. If subtle principles are to be brought to light, they must wait for serendipity. It has been argued that life is complex, it does not yield to few small neat explanations or pigeon-holing, and if there is a unifying principle in biology, it is that there is no unifying principle in biology.

Can ideas from mathematics, computer science, and systems sciences be brought to bear to systematically hunt for principles and patterns that will reveal a grand unified theory of biology? Are their design rules at play in how these systems evolve, interact, and self-assemble? What tools can we steal from other sciences or engineering disciplines to see much more a global view of biology? What can be automated to make computers work on tasks that are humanly impossible? Is systems biology the answer to the problems of biology?

2 Models as Idealization: From Turing to Odell

A particularly revealing example, described below, underlines how biology progresses through both hypotheses and principles—in many false-starts and true-conclusions—as it attempts to develop mathematical models: in this case, models of biological pattern formation.

Alan Turing, an English mathematician, logician, and cryptographer, may be viewed as the last and probably the most neglected proponent of search for principles in biology, despite an immortalized life fueled by an eclectic set of polymathic interests that led to such wide-ranging concepts as Turing machines—the fundamental models of computing, Church-Turing Thesis and Turing reductions—the key ingredients of computability and universality in constructive mathematics, Turing-test of artificial intelligence and Turing patterns arising from reaction-diffusion equations. Turing worked from 1952 until his death in 1954 on mathematical biology, specifically morphogenesis. In 1952, Turing published in Philosophical Transactions of the Royal Society of London [82] one of the most influential and probably the last truly mathematical biology papers: "The Chemical Basis of Morphogenesis." There, Turing proposed reaction-diffusion equations to explain formation of biological patterns, and as the key to understanding developmental biology. His later papers on the subject remained unpublished until 1992 when Collected Works of A.M. Turing was finally printed.

Through this research, Turing was attempting to create "a mathematical model for the growing embryo," as he was intrigued by the process that could self assemble a whole human with a complex thinking brain from a simple fertilized egg. He wished to create a very general program for modelling embryogenesis: Reminiscent of a Newtonian view of science, Turing's 'model' was designed to be "a simplification and an idealization and consequently a falsification." His model was centered on an "abstract" object, which he christened as morphogen, defined as "simply the kind of substance concerned in this theory..." In fact, anything that diffuses into the tissue and "somehow persuades it to develop along different lines from those which would have been followed in its absence" qualifies under this definition.

In this theory, one starts with a variable a, denoting the concentration of some morphogen. The first step would be the one relating the first temporal derivative of the morphogen concentration $\partial a/\partial t$ (its *rate*) to its second spatial derivative $\nabla^2 a$ (its *flux*) through a diffusion constant D_a. The resulting *diffusion equation* is

$$\frac{\partial a}{\partial t} = D_a \nabla^2 a.$$

A much more interesting situation arises when one considers two interacting morphogens with their concentrations denoted by a and b, and also models their interactions through an additional *reaction* part augmenting the diffusion equation:

$$\begin{aligned}\frac{\partial a}{\partial t} &= f(a,b) + D_a \nabla^2 a,\\ \frac{\partial b}{\partial t} &= g(a,b) + D_b \nabla^2 b,\end{aligned}$$

where functions f and g, while nonlinear, could still be fairly simple: e.g., *affine*. For instance, one could have $f(a,b) \equiv a(b-1) - k_1$ and $g(a,b) \equiv -ab + k_2$.

A simple system of *reaction-diffusion equations* such as this can be shown to give rise to a plethora of life-like patterns, that have now come to be known as Turing patterns. Is this how biology of animal development works? Is this how the leopard got his spots? Do we now know how the fly gets its segmentations? What roles do genes play in this? How are they related to morphogens?

Turing viewed that since the role of genes is presumably catalytic, influencing only the rate of reactions, unless one is interested in comparison of organisms, they "may be eliminated from the discussion." His reliance on one grand sweeping principle and the accompanying elegant mathematical models, in the absence of any experimental support, may have been Turing's greatest folly[1]. Developmental genes and their control of "morphogens" would turn out to be extremely important in pattern formation as they began to be discovered through a series of careful (though tedious) genetic knock-out experiments on flies.

Within a year of Turing's seminal paper, Sir Francis Crick and James Watson, both working from Cambridge's Cavendish lab at the time, wrote one of the seminal papers of the last century: "Molecular Structure of Nucleic Acids: a Structure of Deoxyribose Nucleic Acid," (published in the British scientific journal Nature in 1953) [87], and followed it up in a month with another equally important paper entitled "Genetical Implications of the Structure of Doxyribonucleic Acid," [86]. The first of these two papers not only elucidated the double-helix structure of DNA, but much more significantly, also opened the way for a deeper understanding of the most important principles governing biological processes. Watson and Crick, perhaps serendipitously, wrote: "It has not escaped our notice that the specific pairing that we have postulated immediately suggests a possible copying mechanism for the genetic material." DNA, and the genes they contained, inherited, transmitted and used for regulating biological processes, could be seen to be nature's codebooks written in a language that also encoded perhaps all of the fundamental principles of biology. The collection of the genes came to be known as a "genome," and our efforts to read, store, compare, align, decipher, annotate and interpret genomes using massive computation, automation and biotechnological tools became an obsession of our generation. It was felt that if we could "mine" this data, we could learn all the important principles of biology. The debate between "hypothesis-driven science" vs. "data-mining-driven science" would emerge repeatedly in our time, completely oblivious of any historical context.

To a computational biologist, a genome is just a long string made out of just four characters A, T, C, and G (in case of humans, six billion base-pairs encoding the haplotypic genome[2]). This simple one-dimensional object (perhaps with a little auxiliary epigenomic annotation) would encode the genotypes and indirectly, through a Darwinian unidirectional mapping, also the phenotypes. Since evolutionary, relational, and ecological properties, all must depend on either the genotypes directly, or the phenotypes indirectly, all of biological principles must be sought in the genomes, or in the way they are molded by the variation and selection forces. The Crick's central dogma assigned a centrality to the genomes. This dogma (due to Sir Francis Crick in 1958) states that the information flows from DNA to RNA through transcription and then to proteins through translation; these information flows are all unidirectional: "... Once 'information' has passed into protein it cannot get

[1] A computer science colleague, Dr. S. Biswas of IIT-Kanpur, wrote in response to an early draft: "You can see that I'm somewhat unhappy that you say that Turing was wrong– not because of some thing is wrong with your account, but because Turing is Turing!" At some deeper level, there is some truth in this view.

[2] I am counting the base-pairs in **all** your chromosomes, thus including everything you inherited from both of your parents. If I lumped together the homologous sister chromosomes in your diploid genome, the number will be about half: three billion base-pairs encoding the genotypic genome.

out again. The transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein, may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible. Information means here the precise determination of sequence, either of bases in the nucleic acid or of amino acid residues in the protein." Thus once one has performed all the experiments to determine the sequences of the genome, this dogma may suggest to some that all that remains in biology is just data mining.

Also, it may appear that once the genes produce the corresponding proteins through the processes of transcription and translation, they do not play any direct role in the phenotypes determined by protein interaction, except that they have a catalytic role in the rate of production of the proteins. In that case, Turing would have been perfectly right in dismissing the genes as he struggled to create an abstract (hence idealized) view of biology without genes.

However, proteins do interact with DNA, as do many small microRNAs, and in their roles as transcriptional regulators they collude with the genes to affect the dynamics of the entire biological system. Thus the mechanistic picture of biology that one may carry around is the following: A specific region of the genome (DNA) that determines the synthesis of proteins (through the transcription and translation) will be called a gene, thus giving it a physical meaning, even though, originally, a gene meant something more abstract—a unit of rhereditary transmission. Transcription of a gene to a messenger RNA, mRNA, is keyed by a transcriptional activator/factor, which attaches to a promoter (a specific sequence adjacent to the gene). Regulatory sequences such as silencers and enhancers control the rate of transcription. Most of these regulators themselves are proteins synthesized by other genes. These regulators also play a very critical role in the development of an organism and the pattern formation.

The important role of regulators is best appreciated in answering such morphogenetic puzzles as in the elucidation of patterning along the head to tail (antero-posterior) axis of the fruit fly *Drosophila melanogaster*. The circuitry understood from studying a fly applies to other organisms and could reveal some important insights in developmental biology; other multicellular organisms use similar mechanisms for axis formation, and use similar signal transfer between the earliest cells of the developing organisms. The processes important for patterning in the Drosophila embryo are controlled by gradients in the concentration of maternal gene products that arise soon after fertilization. These protein molecules establish broad domains of gene expression that interact to establish final patterns that ultimately encode instructions for body plan (axes: antero-posterior, or dorsal-ventral) or construct germ layers (e.g., ecto-, meso-, endoderm). A simple picture of the processes involved in antero-posterior patterning is described below.

Upon fertilization, various protein (morphogen) concentration gradients spanning the egg are established, starting with the products of the genes, called maternal effect genes, which encode for morphogen proteins. These gradients, in a manner that is generative rather than descriptive, establish positional values and affect the behavior of the other genes. The model now called "PI (Positional Information)" (or more colloquially and colourfully "French-Flag Model") was first proposed by Lewis Wolpert [88], with fairly scant experimental evidence. Wolpert, a civil engineer from South Africa, teaching Cell Biology at King's College in London, proposed in the sixties, that different genes would be activated differently in response to different threshold concentrations of the morphogen. For instance, in order to construct a French flag, nature may start with a linearly arrayed sequence of cells with a morphogen gradient monotonically decreasing from one end to the other, which would activate the cells in the neighbourhood differently: with high concentrations (above an upper threshold) turning on a blue gene, low concentrations (below a lower threshold) turning on a red gene, with white a default state in regions of the embryo at the intermediate concentrations (between the lower and upper thresholds). If one increases the number of cells, or varies the morphogen gradient (without violating the monotonicity property), one still gets the same French flag with surprising robustness. A reader may ponder over the following challenging exercise seeking to devise a similar algorithm to construct an American flag using PI (positional information).

Some of the most important maternal effect genes participating in Drosphila PI model are: Bicoid and hunchback, which pattern anterior parts (head and thorax) of the Drosophila embryo; and Nanos and Caudal, which form posterior abdominal segments of the Drosophila embryo. Bicoid (Bcd) is a homeodomain-containing transcription factor that establishes and places all anterior structures in the Drosophila body plan. After eggs fertilize, bcd (bicoid) mRNA is translated, and a gradient of Bcd (Bicoid) protein is formed, with highest levels near the anterior tip of the embryo, and gradually lower levels toward posterior regions, determined by many parameters: rates of translation, diffusion, and degradation. The experimental evidence supporting Bcd (Bicoid) as a morphogen comes from mutation experiments in which bcd (bicoid gene) is knocked-out or its concentration is shifted, with dramatic effect on the patterns that emerge. For instance, in embryos with an increased copy number of bcd gene (e.g., four or six copies), the cephalic furrow, one of the first distinguishable morphological features, becomes shifted posteriorly.

During the time Bcd protein gradient is forming, zygotic nuclei concomitantly undergo many rapid division cycles and reorganize themselves along the periphery of the embryo. Almost at the same time, the zygotic transcription process gives rise to activation of genes such as hunchback (hb) and orthodenticle (otd), thus turned on by bicoid (bcd). In this way, the first morphogen gradients cascade into other interesting subsequent gradients.

While I omit many of the interesting details, the readers can imagine a rather cleverly regulated process that is at the core of the patterning in biology—a process in which a very specific set of developmental genes play specific and important roles, contradicting Turing's idealization of biology.

In the picture that emerges, the next set of genes to be turned on are called "gap genes." The gradients already established by Bicoid, Hunchback, and Caudal proteins transcriptionally regulate other zygotically expressed protein products, derived from members of the "gap" family of developmental control genes. For instance, hunchback, krüppel, giant, tailless and knirps are all gap genes. These genes, in combination with others, establish the segmentations along the anterior-posterior axis in the body-plan of fly. Thus, the gap genes are usually thought of as the first layer of a hierarchical cascade of the genes that control segmentation.

The next set of genes consists of the so-called Pair rule genes, a class of segmentation genes, expressed after the gap gene products. These genes are expressed in striped patterns of seven bands perpendicular to the anterior-posterior axis. Finally, the ultimate class of segmentation genes, the segment polarity genes, fine-tunes the process by interactions between the cells of adjacent parasegments. A rather elegant example of this process is seen through the reciprocal signaling between Wingless and Hedgehog producing cells, which stabilizes the boundary between each segment.

This tour-de-force of experimental "observational" biological research culminated in 1995 in a Nobel Prize for Physiology or Medicine, awarded to three scientists: Edward B. Lewis, Christiane Nüsslein-Volhard and Eric Wieschaus. This recognition was to be seen as a testimony to the power of careful experiments based on genetic screening of embryo patterning mutants that revealed the role, played in early embryologic development by Homeobox genes like bicoid. Perhaps, after all, Hooke had won the argument. It is entirely possible that the theoretical elegance of abstraction of the kind Turing was proposing is rather seductively misleading; truth must be sought in the details, no matter how confusingly ugly.

But, our viewpoints shifted again by subsequent computational research reported in an article that appeared in July 2000 issue of *Nature*, where von Dassow et al. [85] reported a mathematical analysis of a model based on the earlier experimental work that I have just described. The authors of this article used a computer simulation to examine the robustness of a simple working model of patterning in fly. This computational model, based upon a kinetic mass action formulation of the biochemical reactions, consisted of a system of coupled differential algebraic equations, derived from two categories of first-order ordinary differential equations: namely, equations characterizing the transcriptional processes, and equations characterizing translational processes. In this formulation, the rate of change of mRNA product concentration is expressed by combining two terms: an additive (positive) term accounting for the way the transcriptional regulators synthesize the mRNA and a subtractive (negative) term accounting for the way the mRNA degrades or is removed. The first term is expressed by a rational function involving polynomials whose degrees are determined by various cooperative interactions among the regulators (Hill's coefficients in Hill-type equations), and the second term is a linear function. They all involve many free parameters, whose values are often not known a priori. Similarly, the rate of change of protein product concentration is expressed by combining two linear terms: an additive term for protein synthesis through translation and a subtractive term for protein degradation. However, their first model, which had encoded faithfully the accepted interactions based on the experimental work, failed, and upon debugging, found to require the addition of some interactions which could be only vaguely substantiated. The revised model of fly segment polarity, which deviated from the experimental evidence reported so far, turned out to be highly robust, insensitive to variation in parameters and initial conditions. The revised model, developed by Odell's group, encompassed 136 coupled equations and nearly 50 free parameters such as half-lives, diffusion constants and binding coefficients, but despite this complexity, the model exhibited unusual robustness. For instance, even when the authors picked random values for the free parameters (ranging over a wide interval varying over several order of magnitudes, but still within a generally realistic range), they discovered that the revised model still exhibited the patterns seen in real flies.

After having exhausted various mathematical, experimental and computational tools available, one may still wonder whether it is possible to be reasonably convinced of the finality of this answer(s)—even when the question is narrowed down to just the topic of biological patterning. *Is this our final answer*?

As this example illustrates, this issue of truth and cognitive finality in biology remains a rather unsettled and unsettling topic. At present, it is not uncommon to assume many biological problems to have been satisfactorily solved without rigorous justification. Thus, it is hoped that rigorous mathematical models with automated tools for reasoning, simulation, and computation can be of enormous help to uncover cognitive flaws, qualitative simplification or overly generalized assumptions. Some ideal candidates for such study would include: prion hypothesis, cell cycle machinery, muscle contractility, most of the processes involved in cancer (cell cycle regulation, angiogenesis, DNA repair, apoptosis, cellular senescence, tissue space modelling enzymes, etc.), signal transduction pathways, and many others. The story above brings us to our next topic: What role can Systems Biology play in better, rigorous and refined elucidation of biological principles?

3 Model Building and Model Checking: Kripke and Tarski

Systems Biology is characterized as a broad and challenging field attempting to hasten the understanding of life by integrating computational, mathematical, statistical, artificial intelligence, machine learning, temporal reasoning, and other so-called *in silico* techniques coupled with traditional wet-lab *in vivo* and *in vitro* experimentation. As illustrated earlier, one important strand of research has been modelling, simulation and analysis of biochemical pathways to evaluate mutual consistency of biological facts, to validate or refute plausible hypotheses, to aid experiment design and refine existing models. *One may then safely posit that the fundamental systems biology problem addressing biochemical networks will continue to be reasoning about the emergent temporal properties and their modulation by external signals.* However, as we saw earlier, the direct detailed kinetic mass action based numerical analysis of biochemical networks is simultaneously limited in global analysis and intractable in large-scale exploration, because of the complexity of the network in terms of the unknown parameters and initial conditions, the number of interacting species, the number of interconnections and the different types of interactions. More importantly, it becomes extremely difficult to reason about the system-level properties of the entire network from the time-course analysis of its many molecular components at several different initial conditions and kinetic parameter values. The kind of debugging that Odell's group performed on their model had to depend on human ingenuity.

Nevertheless, understanding biological models well and developing the abilities to manipulate them have important payoffs—both in practice and theory. A systems level analysis (as opposed to a reductionist approach) of biology is crucial not only for giant leaps in our understanding of "plausibility of life" itself [32], but also for more mundane applications to vaccine and drug discovery, diagnostics, genetically modified food, and in the nascent field of synthetic biology [22, 31]. The data generated by the recently announced American National Cancer Institute's TCGA (The Cancer Genome Atlas) project, when combined with proper systems biology tools modelling signal transduction pathways, can be hoped to replicate the success of cancer drugs like Her 2 and Gleevec (Imatinib Mesylate), and shed light on why other cancer drugs like Iressa (gefitinib) and Tarceva (erlotinib) have only had limited success in few populations.

One is further inspired by the techniques of model checking, a popular computational approach, which has had many important impacts [71, 25] in engineering fields because of its ability to efficiently validate temporal logic formulæ, describing dynamical properties of the formal models of complex real systems. These well-established techniques have been tremendously successful in VLSI circuit design, robotics, control theory, etc., and have been applied even to hybrid systems with both continuous and discrete dynamics. For biochemical networks, several modelling tools, some with rigorous analytical capabilities, have been developed, including Systems Biology Workbench [46], Virtual Cell [77], E-Cell [79], Genomic Object Net [68], Simpathica [9] and BioMiner [76]. Unfortunately, many of these currently available tools are either rudimentary and do not scale, or are based on numerical simulation and animation, and hence can provide only limited insight with results that are difficult to generalize. There is a dearth of symbolic[3] techniques for analyzing the dynamics of biochemical systems—in particular, a characterization of the functional relationship between the initial conditions and the emergent phenomena. In addition to yielding generic results, symbolic methods result in improved efficiency as computations do not have to be redone when initial conditions or parameters change.

One must then strive to develop sophisticated parallelizable dense-time algebraic model checkers that can be used by biologists, bio-technologists, et al. to reason about and manipulate important biological processes through richer and deeper symbolic queries. Concurrently, one may seek to elicit a better understanding of complexity and decidability [13] issues in algebraic model checking. These ideas are

[3] In this paper, "symbolic" and "algebraic" denote the retaining of variables and parameters in their symbolic forms rather than as numbers, and are not to be confused with symbolic model checking using binary decision diagrams.

elegantly integrated under the umbrella of "Algorithmic Algebraic Model Checking"(AAMC) [69, 67, 65, 20, 21, 19, 64, 66, 63, 16].

3.1 Problem Domain

We start with the following taxonomy into which the cellular biochemical processes are typically organized, as described below.

Genetic Regulation: The oft-repeated "central dogma of biology" states that biochemical information in cells is encoded primarily in the Deoxyribo Nucleic Acid (DNA) molecules. DNA is *transcribed* into messenger Ribo Nucleic Acid (mRNA), and the mRNA then is *translated* into proteins at the ribosomes. Genetic regulation is the process of modulation of the expression of the relevant genes at the correct locations and times, and is keyed by specific proteins called transcriptional factors. Through transcriptional factors and other ancillary modulators, proteins, the products of genes, themselves partake in this genetic regulatory process, thus giving rise to complex interaction networks; such proteins interact with regions of the DNA to effect modulation of how genes are transcribed. The binding of the transcription machinery and the transcriptional factors to the DNA involves complex protein-DNA-protein interactions, where, more often than not, the structural modification of the DNA (such as euchromatin and heterochromatin regions) and the protein has to be accounted for.

The rate of gene transcription, the post-transcriptional mechanisms that affect mRNA half-life (i.e., stability) and the formation of the mRNA-ribosome complex are other aspects of genetic regulation. Similarly, there are post-translational mechanisms for protein modification such as phosphorylation of key residues, multimerization, chaperone-guided complex formation, protein-folding control, and genetic control by small interfering RNA (siRNA), all of which must be faithfully included.

Signal Transduction: The cell responds to external signals through receptors, which may be on its surface or in its cytoplasm. The signal is transmitted to the interior through messengers, which induce the desired response to the external signal. Typically, a ligand binds to a trans-membrane receptor whose conformation subsequently changes. This change is detected by proteins bound to it (usually on the cytoplasmic side), or is manifested as a change in the receptor's chemical properties. Subsequently, second messenger molecules amplify the signal and communicate it to the target(s). Alternatively, the ligand can directly enter the cell through non-specific channels and then bind to the receptors inside the cell. Small molecules like calcium often participate in these pathways, where most of the reactants are enzymatic proteins. The net result of the signal transduction pathway is an appropriate response by the specific subcellular component. Very often, the signaling pathway results in the nuclear localization of transcriptional factors, leading to the transcription (or shutting down) of corresponding genes. The binding of the signaling molecule with the receptor, the modification of the structure of the receptor and associated proteins (with the receptor sometimes acting as an enzyme) and dispatching of second messengers are the activities near the cell membrane. Receptor desensitization, internalization and regeneration are other complex sub-processes, thus altering the physical properties of binding and diffusion.

Metabolism: Metabolism represents almost all processes that are not genetic regulatory or signal transducing. The gigantic set of biochemicals needed by the cell are continuously produced and consumed by complex enzyme catalyzed pathways. These comprise the metabolic network. They essentially govern the matter and energy cycles of a cell— the way energy and matter are obtained, transformed and consumed by living organisms. Photosynthesis for example is the process by which light energy is converted into chemical energy during sugar (e.g., glucose) formation. During respiration, the oxidation of glucose transforms the energy into Adenosine Tri-Phosphate (ATP). While the ATP-cycle and photosynthesis comprise the well-known energy metabolism, carbohydrate metabolism deals with Glycolysis and Phosphates, lipid metabolism pertains to Triacyl Glycerol and Fatty Acids, and amino acid metabolism mostly refers to Glutamate and Urea.

Other Processes: Biology is complex, and of course, there are still more aspects to cellular biology beyond this simple trichotomic characterization. These include the biophysics of DNA packaging, small interfering RNA (siRNA), protein folding and DNA-protein interaction, cell adhesion, non-transcriptional regulatory pathways, cellular compartments and related spatio-temporal phenomena, cell proliferation, and cell migration. While the modelling approaches suggested here, when further augmented with suitable stochastic and spatial formalisms, will generalize as well, I will not emphasize those applications directly in my discussions here.

3.2 Biochemical Models

The different component parts and processes in the biochemical domain may be represented at different levels of abstraction [33, 47]. I summarize some of the major approaches below, but will guide the discussion towards hybrid automata representation, a very general and powerful model for these systems.

Logical Modelling: The state of the reactant is captured through a finite number of abstract-states (where intermediate expression levels are assumed to have the same behavior), and functions are used to describe the new states (concentration range) of the chemical species, given their old states. The transitions between states can be assumed to occur synchronously or (more accurately) asynchronously. In the simplest case, only two states ("on" and "off") are used, and Boolean algebra is used to describe the dynamics. Literature on Concurrent Transition Systems [24, 23] and Pathway (Rewrite) Logic [36] provides good expositions of logical modelling. Kappler et al. [51] demonstrate how to extend simple Boolean networks by using ordinary differential equations to capture the concentration, while Boolean functions continue to determine the rates of the reactions. The probability of being in a state is sometimes a more reasonable measure to estimate, as in the case of Sachs et al. [74], who use Bayesian networks to model cell signaling pathways. Similarly, Shmulevich et al. [75] describe the use of probabilistic Boolean networks to model genetic regulatory networks and determine the long-term joint probabilistic behavior of a few selected genes. Platzer et al. [70] simulate the embryonic development of *C. elegans* by assuming Boolean states for the genes and synchronously updating at each time step based on an interaction matrix. Batt et al. [11] have applied model checking theory on biochemical systems modeled though qualitative simulation.

Differential Equations: If instead the concentrations are represented exactly in the real continuous domain, the ordinary differential equations (ODEs) of the dynamics directly follow from the law of general mass action (GMA) [30, 52, 83]. For instance, in the reaction $aA + bB \longleftrightarrow cC + dD$, the rate of the forward reaction $v_f \equiv k_f[A]^a[B]^b$ and the rate of the backward reaction $v_b \equiv k_b[C]^c[D]^d$, where k_f and k_b are the forward and backward rate constants respectively and the rate of individual reactants is $\frac{1}{c}\dot{C} = \frac{1}{d}\dot{D} = -\frac{1}{a}\dot{A} = -\frac{1}{b}\dot{B} = (v_f - v_b)$. As a compromise between discrete and continuous representations, qualitative differential equations can be used with qualitative states corresponding to the different concentration ranges [11, 34]. Models, involving spatially distributed or probabilistic interactions among molecules, require several other mathematical generalizations, e.g., partial differential equations (PDEs), stochastic differential equations (SDEs), or reaction-diffusion equations.

Hybrid Systems: Many biological systems, such as the cell, follow a combination of discrete and continuous behaviors, which cannot be characterized in a proper way using either only discrete or only continuous models. On one hand, their evolution is ruled by a continuous dynamical law concerning substance concentrations and gradients, and, on the other hand, such a dynamical law may change discretely depending on the system status itself. Because of their hybrid nature, part discrete and part continuous, such systems are named hybrid systems. To model hybrid systems, Alur et al. introduced the notion of hybrid automata in [3]. Intuitively a hybrid automaton is a "finite-state" automaton with continuous variables, which evolve according to a set of continuous laws characterizing each discrete mode of the automaton itself. The use of hybrid automata for modelling biomolecular networks has been described by Alur et al. [1] and Mishra et al. [61]. Amonlirdviman et al. [6] demonstrated the utility of hybrid systems by modelling Drosophila planar cell polarity. Starting with the S-System formulation of Savageau and Voit [84], Antoniotti et al. [8] used an additional automaton to broaden the set of representable systems, subsequently using full-fledged hybrid automata [7]. Ghosh et al. presented both delta-notch [39, 38] and protein signaling network [40] models based on the hybrid automaton formalism. Casagrande et al. [19] suggested a simple (and decidable) hybrid automaton model for the *E. coli* chemotaxis (see the example later). Lincoln and Tiwari [57] detail hybrid automaton modelling of biochemical networks, while Hu et al. [45] describe stochastic hybrid system modelling of subtilin production in *Bacillus subtilis*. More recently, Drulhe et al. [35] have described piecewise-affine models of genetic regulatory networks.

Algebraic Hybrid Automata, Dense-time Logic and Decidability: Thus, if these methodologies are to be adapted to the settings suitable for systems biology, what is needed is an appropriate generalization of discrete-time systems, classical temporal logic, possible-world models of temporal logic given by Kripke (e.g., Kripke structures), model checking algorithms based on graph theoretic analysis, etc. However, while the generalization must be sufficiently powerful to capture a large segment of reasoning processes used by the biologists, but yet it should also be sufficiently constrained so that these systems can be reasoned by feasible computational means. At the least, the resulting problems should be decidable (computable). We seek such a framework below by a judicious amalgamation of symbolic algebra (e.g., using decision procedures of semi-algebraic geometry), sufficiently constrained dense-time temporal logic and algebraic models based hybrid automata. I will present a brief formal discussion of such hybrid automata and their reachability problem, and then discuss an appropriate dense-time logic and the complexity of the model checking problem in this framework. The discussion, with few accompanying biological examples, indicates how close these techniques are to the boundary of undecidability.

4 Algorithmic Algebraic Model Checking

My research group introduced the subject of *Algorithmic Algebraic Model Checking* in an attempt to rigorously examine connections between systems biology, dynamical systems, modal logic and computability, and how they can be useful in the biological context. Towards this aim, we started by addressing the symbolic model checking problem for a new class of hybrid models arising in systems biology – *semi-algebraic hybrid systems*, introduced in the first paper of our "AAMC" (Algorithmic Algebraic Model Checking) series [69]. In there, and its subsequent sequels, we sought to characterize the widest range of automata that admit sound albeit expensive mathematical techniques, as opposed to focusing on a very narrow class of systems that often prematurely sacrifice genralizability for the sake of efficiency. From this research, a new class of automata, IDA (*Independent Dynamics Hybrid Automata*), has emerged as one that satisfies the desiderata of systems biology.

In preparation for these systems biology models, I start by providing a formal definition of *hybrid automata* — a concept first introduced in [59, 4] as a model and specification language for hybrid systems, i.e., systems consisting of a discrete program within a continuously changing environment.

4.1 Syntactical Structure

First, some notations and conventions need to be introduced. If $p = (p_1, \ldots, p_k)$ and $s = (s_1, \ldots, s_k)$ are vectors in $\mathbb{R}^k$, $r \in \mathbb{R}_{\geq 0}$, $\mp \in \{-, +\}$, and $\asymp \in \{\leq, <, =, >, \geq\}$, then I will use $p \mp s$ to denote the vector $(p_1 \mp s_1, \ldots, p_k \mp s_k)$ and $\|s\| \asymp r$ to indicate equational relations such as $\left(s_1^2 + \ldots + s_k^2\right) \asymp r^2$.

Indexed capital letter variables Z_m, and Z'_m, where $m \in \mathbb{N}$, denote variables ranging over $\mathbb{R}$, while Z, and $Z^{'}$ denote vectors of variables $(Z_1, \ldots, Z_k)$, and $(Z'_1, \ldots, Z'_k)$, respectively.

The temporal variable T models time and range over $\mathbb{R}_{\geq 0}$. In the following, given a formula $\psi[Z]$ and a model $\mathcal{M}$, I will denote the set of tuple of values satisfying ψ in $\mathcal{M}$ as $Sat(\mathcal{M}, \psi)$, i.e., $Sat(\mathcal{M}, \psi) \overset{\mathrm{def}}{=} \{p \mid \mathcal{M} \models \psi[p]\}$. When $\mathcal{M}$ is clear from the context, I may simply write $Sat(\psi)$.

A formal definition of hybrid automata may proceed as follows: For each state of a discrete automaton there is an *invariant condition* as well as a *dynamic law*. This dynamic law may depend on the initial conditions, i.e., on the values of the continuous variables at the beginning of the evolution in the state. The discrete jumps from one discrete state to another are regulated by the so-called *activation* and *reset* conditions. As an example, one may imagine a two-state hybrid automaton with the states $\{q_1, q_2\}$, with activation conditions from q_1 to q_2 (respectively, q_2 to q_1) given by inequality relations $G_{12}(x) \geq 0$ (respectively, $G_{21}(x) \geq 0$). Furthermore, as the automaton enters these states, it resets itself to some values, say x_1 in the state q_1 and x_2 in state q_2. Starting from this skeletal description of the automaton's "discrete" behavior, the complete description augments it with more details about what transpires between the discrete switches: namely, how the automaton evolves in each of these states according to a continuous dynamics that is then described by flow and invariant functions: namely, $\dot{x} = f_i(x)$ and $g_i(x) > 0$ for state q_i. Thus this automaton may start in state q_1 with a value x_1 and evolve continuously for a period according to the local flow described by $f_1(x)$, while constrained to the subspace whose boundary is defined by $g_1(x)$, until the activation condition $G_{12}(x) \geq 0$ triggers an instantaneous jump to state q_2. In the new state, it continues according to different flow and invariant rules for the next period before jumping back to q_1. A classical example of a hybrid automaton may be provided by a thermostat that maintains a room temperature in the range [15°C, 20°C]. The thermostat switches back and forth between two states q_{off} and q_{on} depending on whether the room is cold or hot. If the room is cold ($x < 15°C$) then the heat must be turned on, and the temperature will increase according to the law $\dot{x} = k_h - k_r x$. On the other hand, If the room is hot ($x > 20°C$) then the heat must be turned off, and the temperature will decrease according to the law $\dot{x} = -k_r x$. The reset and invariant conditions are the obvious ones. I will next describe these same ideas bit more formally.

Let $\mathcal{L}$ be a first-order language over the reals and $\mathcal{M}$ be a model of $\mathcal{L}$. In the following definition, it will be implicitly assumed that *Inv*, *Dyn*, *Act* and *Reset* are formulæ of this first-order language $\mathcal{L}$.

Definition 4.1 (Hybrid Automaton) *A hybrid automaton (of dimension k)* $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset\rangle$ *(over* $\mathcal{M}$*), consists of the following components:*

1. *$Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals, $\mathbb{R}$;*
2. *$\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph; the vertices of $\mathcal{V}$ are called* locations*, or* control modes*, the directed edges in $\mathcal{E}$,* control switches*;*
3. *Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds (in $\mathcal{M}$), then $Dyn(v)[p, p, 0]$ holds as well;*
4. *Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.* □

The formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ are said *invariant* of v and *dynamics* of v, respectively, while $Act(e)[Z]$ and $Reset(e)[Z, Z']$ are called *activation* of e and *reset* of e, respectively. Moreover, if a reset does not depend on Z, then it is said to be a *constant reset*.

From such formulæ one can define the formula $\overline{Reset}(e)[Z] \overset{\text{def}}{=} \exists Z'\, Inv(v)[Z'] \wedge Act(e)[Z'] \wedge Reset(e)[Z', Z] \wedge Inv(u)[Z]$, where $e = \langle v, u \rangle$.

Henceforth, one may simply write $\mathcal{A}(e)$, and $\mathcal{R}(e)$ to mean $Sat(Act(e))$, and $Sat(\overline{Reset}(e))$, respectively.

In the above definition, instead of using the classical approach based on differential equations to define the flow, one may use the formulæ in *DynSet* to describe the continuous evolution without using derivatives. (See [14] and also [56]). This simplification allows such hybrid automata to generalize several recently discovered notions in this theory. For instance, this step allows treatment of o-minimal hybrid automata [56, 14] as a special case of the resulting automata, and map rectangular hybrid automata [72, 41, 54] into a subclass of the earlier definition.)

On occasion, one may wish to simply express hybrid automaton dynamics using differential expressions (either equations or inclusions). Let $\mathscr{R}$ be a function assigning to each vertex $v \in \mathcal{V}$ a system of differential inclusions (that can become a system of differential equations, as a particular case). I will occasionally use the notation $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, \mathscr{R}, Act, Reset \rangle$ instead of $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ to denote the fact that, for each vertex $v \in \mathcal{V}$, the formula $Dyn(v)[Z, Z', T]$ corresponds to the solution of the differential inclusions $\mathscr{R}(v)$ when the starting point is Z.

4.2 Semantic Structure: Reachability and Beyond

A suitable formalization of the semantics of hybrid automata is built upon the concepts of hybrid automaton's states and trajectories (traces).

Definition 4.2 (States) *Let H be a hybrid automaton (over $\mathcal{M}$) of dimension k. A* state *q of H is a pair $\langle v, r \rangle$, where $v \in \mathcal{V}$ is a location and $r = (r_1, \ldots, r_k) \in \mathbb{R}^k$ is an assignment of values for the variables of Z. A state $\langle v, r \rangle$ is said to be* admissible*, if $Inv(v)[r]$ holds in $\mathcal{M}$.* □

Dynamics of a hybrid automaton can then be described via sequences of transitions from one state of the automaton to another. Hybrid automata have two kinds of transition (and reachability) relations: *continuous transition (reachability) relations*, capturing the continuous evolution of a state according to both formulæ $Dyn(v)$ and $Inv(v)$, and *discrete transition (reachability) relation*, capturing changes of location driven by the formulæ $Reset(e)$ and $Act(e)$. More formally, the semantics of our hybrid automata is described as follows.

Definition 4.3 (Hybrid Automaton - Semantics) *Let H be a hybrid automaton (over $\mathcal{M}$) of dimension k. The* continuous reachability transition relations $\xrightarrow{t}_C$ *between admissible states is defined as:*

$$\begin{aligned} &\langle v, r \rangle \xrightarrow{t}_C \langle v, s \rangle \iff \\ &\quad \textit{there exists } f : \mathbb{R}_{\geq 0} \to \mathbb{R}^k \textit{ continuous function} \\ &\quad \textit{such that } r = f(0),\ s = f(t), \textit{ and for each } t' \in [0, t] \\ &\quad \textit{the formulæ } Inv(v)[f(t')] \textit{ and } Dyn(v)[r, f(t'), t'] \textit{ hold} \\ &\quad \textit{in } \mathcal{M}.\ f \textit{ is called } \text{flow function}. \end{aligned}$$

The discrete reachability transition relation $\xrightarrow{e}_D$*, where $e \in \mathcal{E}$, between admissible states is similarly defined as:*

$$\begin{aligned} &\langle v, r \rangle \xrightarrow{\langle v, u \rangle}_D \langle u, s \rangle \iff \\ &\quad \langle v, u \rangle \in \mathcal{E} \textit{ and the formulæ } Inv(v)[r],\ Act(\langle v, u \rangle)[r], \\ &\quad Reset(\langle v, u \rangle)[r, s], \textit{ and } Inv(u)[s] \textit{ hold in } \mathcal{M}. \ \square \end{aligned}$$

I will also use the notation $\ell \xrightarrow{\lambda} \ell'$ to indicate that either $\ell \xrightarrow{\lambda}_C \ell'$, if $\lambda \in \mathbb{R}_{\geq 0}$, or $\ell \xrightarrow{\lambda}_D \ell'$, when $\lambda \in \mathcal{E}$.

Example 1 Let H be a hybrid automaton with $\mathcal{V} = \{v\}$, $\mathcal{E} = \{\langle v, v \rangle\}$, and in which $Dyn(v)[Z, Z', T]$ is $Z' = e^T * Z$, $Inv(v)[Z]$ is $1 \leq Z \leq e^2$, $Reset(e)[Z, Z']$ is $Z' = 1$, and $Act(e)[Z]$ is $4 \leq Z \leq e^2$. One may choose the following transition sequence tr as valid for H: $\langle v, 1 \rangle \xrightarrow{2}_C \langle v, e^2 \rangle \xrightarrow{\langle v, v \rangle}_D \langle v, 1 \rangle$. If, now, one defines another automaton H', by modifying only the invariant formula $Inv(v)[Z]$ of H to change to $1 \leq Z < e^2$, then by the semantics proposed in [44, 58], tr is still valid for H', while it is no longer valid by our semantics. Note that $\langle v, e^2 \rangle$ is not a valid state since $Inv(v)[e^2]$ is now false, and is not admitted in a trace in the way I have defined it. Except for few subtle differences such as this, I have otherwise closely followed the classical definition through out this paper. □

Without loss of generality, I focus only on hybrid automata whose formulæ are satisfiable. For this reason, one can safely omit any mention of the model over which the automaton is constructed or to the dimension of the automaton, except when it is ambiguous in the context.

Definition 4.4 (Trace) *Let H be a hybrid automaton, ℓ be an admissible state of H, and let $J \subseteq \mathbb{N}$ be an initial segment of $\mathbb{N}$ ($|J| > 1$).*

A trace *of H is a sequence $(\ell_j)_{j \in J}$ of admissible states such that:*

1. *for all $j \in J \setminus \{0\}$ $\ell_{j-1} \xrightarrow{\lambda} \ell_j$;*
2. *for all $j \in J \setminus \{0, 1\}$ there exist e in $\mathcal{E}$ such that either $\ell_{j-2} \xrightarrow{\lambda} \ell_{j-1} \xrightarrow{e}_D \ell_j$, or $\ell_{j-2} \xrightarrow{e}_D \ell_{j-1} \xrightarrow{\lambda} \ell_j$,*

where $\lambda \in \mathcal{E} \cup \mathbb{R}_{\geq 0}$. □

At this point, a couple of remarks are in order:

Remark 4.1 Condition 2 in the above definition admits a notion of hybrid trace analogous to the notion of trajectory defined in dynamical systems; without such a condition, a transitive dynamics must be assumed. Consider, for instance, the dynamics $Dyn(v)[Z, Z', T]$ defined as $Z'_1 = T + Z_1 \wedge Z'_2 = T^2 + Z_2$, with $Z = (Z_1, Z_2)$ and $Z' = (Z'_1, Z'_2)$. Intuitively, one does not expect to admit a trajectory from $(0, 0)$ passing through $(4, 8)$, satisfying Dyn. However, without Condition 2, $\langle v, (0, 0) \rangle \xrightarrow{2}_C \langle v, (2, 4) \rangle \xrightarrow{2}_C \langle v, (4, 8) \rangle$ would be a legal trace from $\langle v, (0, 0) \rangle$ passing through $\langle v, (4, 8) \rangle$. □

Remark 4.2 There can exist traces which do not spend much time in continuous evolution and, in fact, time does not even advance on them. Hybrid automata admitting such traces are called *Zeno* hybrid automata. Since allowing such systems in defining models of systems biology may lead to counter-intuitive conclusions, special care is called for in such situations. □

I will now formally describe the notion of *reachability.*

Definition 4.5 (Hybrid Automaton - Reachability) *Let H be a hybrid automaton of dimension k. A point $r \in \mathbb{R}^k$* reaches *a point $s \in \mathbb{R}^k$ (in time t) if there exists a trace $tr = \langle v, r \rangle, \ldots, \langle u, s \rangle$, for some $v, u \in \mathcal{V}$ (and t is simply the sum of the elapsed times in continuous transitions).*

We use $ReachSet(r)$ to denote the set of points reachable from r. Moreover, given a region $R \subseteq \mathbb{R}^k$ we use $ReachSet(R)$ to denote the set $\cup_{r \in R} ReachSet(r)$. □

A naïve approach to compute reachability relation may involve simply iterating over the computation of points locally reachable through continuous and discrete transitions. Unfortunately, this procedure is not *effective* in general, as it immediately faces two difficulties: namely, transitions might be characterizable only by undecidable formulæ; or even when single transitions are computable, the global procedure may not be guaranteed termination. Other attempts, for instance, the following fix-point result, defined in terms of certain computable "forward time closure operators," $\langle \cdot \rangle^{\rightarrow}$ and $[\cdot]^{\rightarrow}$, and presented in [3] by Alur et al., suggest computable algorithms for reachability, only in certain simple situations.

Proposition 4.1 *Let H be a hybrid automaton and R be a H's region. The reachable region $ReachSet([R]_{\Downarrow})$ is least fix point of the equation:*

$$X = \left\langle R \cup \bigcup_{e \in \mathcal{E}} [X]_e^{\rightarrow} \right\rangle^{\rightarrow}. \quad \Box$$

More generally, it has been shown in [42] that reachability is not decidable. Nonetheless, many interesting classes of hybrid automata over which reachability is decidable have been characterized in the literature [54, 41, 56, 14]. A common approach for deciding reachability of hybrid automata employs the technique of discretising the automata either using equivalence relations which strongly preserve reachability (e.g., bisimulation [56]) or using abstractions (e.g., predicate abstraction [5, 81]). The classes of hybrid automata that owe their decidability to bisimulation include the following: *timed automata*, certain subclasses of *rectangular automata*, and *o-minimal automata*, and are defined as follows.

Both timed and rectangular automata are special cases of linear hybrid automata, which is not immediately useful as high-fidelity models in systems biology. For a linear hybrid automata, its dynamics, invariants, and activation relations are all defined by linear expressions over the set Z of variables. For the control modes the dynamics is defined by a differential equation of the form $\dot{z} = k$, where k is a constant, one for each variable in Z, and the invariants are defined by linear equalities and inequalities (corresponding to a convex polyhedron) in Z. Also, for each transition, the set of reset assignments consists of linear formulæ in Z, too. For automata of these kinds a trace is a piecewise linear function whose values at the points of discontinuity are finite sequences of discrete changes. A particularly well-studied, but somewhat artificial, special case of a linear hybrid automaton is a timed automaton, in which each continuous variable increases uniformly with time ($\dot{z} = 1$) and models clocks in asynchronous or self-timed systems. A discrete transition may either reset or disregard the clock.

A slightly more interesting specialization of a linear hybrid automaton, that was mentioned earlier, is a rectangular automaton. A hybrid automaton is rectangular, if the continuous dynamics are independent of the control modes and the variables are pairwise independent. Here, the dynamics is described by differential inclusions of the form $\dot{z} = [a, b]$ for each variable. The invariant condition and the transition relation are described by linear predicates that correspond to hyper-rectangles. For rectangular automata, the reachability problem (also the more useful controller synthesis problem) is decidable, when the mode switches can be assumed to occur through discrete/integral sampling.

O-minimal systems generalize the systems of hybrid automata considerably, by allowing much more general classes of function to describe the dynamics, invariants, and activation, but constrain the reset operation to constant values. The decidability has been shown for a very general (somewhat technical) class, describable in the o-minimal theories, whose definable sets are finite unions of points and intervals (with respect to a linear order relation over its models). In spite of the technical nature of these definitions, these automata present a glimpse of what could be attempted in systems biology models, where discrete mode switches and differential algebraic equations abound. However, one disappointing limitation (at least from the viewpoint of biological modelling) is the requirement of constant-reset.

Although at first glance, it may appear that such restrictions must necessarily be imposed, if one wishes to achieve decidability (especially, using bisimulation techniques), fortunately, there are still other algorithmic approaches to explore. For instance, several general and novel techniques have emerged from the themes of the AAMC-program, in which computational problems on hybrid automata can be fruitfully studied via translation of the problems, such as reachability, into satisfiability of first-order formulæ over decidable theories. In particular, one can make use of the following results (proof omitted).

Theorem 4.1 *If a class $\mathcal{H}$ of hybrid automata is first-order definable and its underlying $\mathcal{T}(\mathcal{M})$ is decidable, then the membership problem for a given hybrid automata H is decidable.*

Furthermore, if the reachability problem for a given hybrid automata H is first-order definable and $\mathcal{T}(\mathcal{M})$ is decidable, then the reachability problem for H is decidable. □

As explained earlier, the ubiquitous tussle between complexity of the system and the ease of analysis continues to dictate which hybrid automaton subclass will ultimately prevail as most appropriate for which application. For example, though remarkably efficient verification techniques have been perfected for Timed Automata [3], the very idea of modelling a biochemical system as a set of well-behaved clocks is largely untenable. At the other end of the spectrum, detailed spatio-temporal models at the atomic level, though perhaps extremely accurate, will leave simulation-based analysis as the only feasible option.

In [20, 21], we introduced the *Semi-Algebraic Constant reset Hybrid Automata* (SACoRe), which extended O-minimal automata over the reals, in the case of flows obtained from non-autonomous systems of differential inclusions. SACoRe automata were shown to admit decision procedures for reachability and model checking for a limited fragment of CTL, which they achieved by combining Tarski's decidability result over the reals with Michael's selection theorem. However, this formalism is still quite restrictive in the biochemical domain as the constant reset requirement that they impose is very limiting. This constraint is rather artificial in systems biology since, when a biochemical system changes its "discrete" state, it would be rather unnatural to assume that the concentrations are reset to constant values. In fact, the most common result of a state change is no perceptible transient variation in continuous variables, because of the underlying thermodynamic model built upon large number of molecular interactions over a short time interval. In other words, identity resets are unavoidable, if one wishes to capture such fundamental aspects of biological state transitions.

For this purpose was introduced a new class of hybrid automata – *Independent Dynamics Hybrid Automata* (IDA), whose characterizing conditions are based upon a decidable first-order theory over the reals (e.g., ($\mathbb{R}$, 0, 1, +, ∗, =, <)). The dynamics are solutions of autonomous systems of differential equations. The reset conditions can be either constants as in the case of O-minimal hybrid automata [56] or the identity function. In particular, these models distinguish *independent variables*, whose resets are the identity function, from *dependent variables* whose resets are constant functions. The flows and the reset functions of the dependent variables are allowed to depend on the independent ones, but not *vice versa*.

In the most general setting, one cannot impose any bound on the time interval over which an IDA's dynamics can be satisfactorily examined, but one is able to define such bounds[4] for an interesting subclass of IDA, called ∞IDA. As

[4] Most importantly, observe that it is not necessary to explicitly compute these time bounds, but only to check their existence, again,

a consequence, reachability is always decidable on ∞IDA. In this manner, IDA automata achieve following two desiderata: (1) They extend O-minimal automata to make them suitable for Systems Biology applications; (2) They restrain Semi-Algebraic Hybrid Automata (defined in the first AAMC paper) and make them more amenable to analysis. IDA automata accomplish these by exploiting the decidability of the first-order theory over which it is defined, and thus can bound the time interval that must be considered to solve a reachability problem as well as prove the decidability of reachability. When the IDAs of interest are defined on the first-order theory ($\mathbb{R}$, 0, 1, +, $*$, =, <), our approach exploits Tarski's result and quantifier elimination to characterize reachability.

The reachability problem for hybrid automata is now formulated as follows: given a hybrid automaton H and two formulæ, ι and τ, denoting an initial set of points $Sat(\iota) \subseteq \mathbb{R}^k$ and a target set of points $Sat(\tau) \subseteq \mathbb{R}^k$ respectively, one desires to decide whether there exists a point in $Sat(\iota)$ which reaches a point in $Sat(\tau)$.

In the IDA class of hybrid automata, the components of Z can be partitioned into two sets: the *independent* and the *dependent* variables. I will denote by X the vector of independent variables which maintains the same component ordering of Z. Similarly I will indicate with Y the vector of dependent variables and with X' and Y' the primed version of X and Y respectively. The independent variables are never reset and their dynamics are the same in all the locations. This condition is similar to that used in rectangular initialized hybrid automata (see [41, 54]).

Moreover, one needs to impose conditions that will ensure the existence of a minimum amount of time, which has to be spent in a location between two jumps. In particular, I will require that the invariants are closed and bounded, and that the distance between reset and activation regions assumes a positive value.

In condition 3 of the definition, I will need that the independent variables which are not reset have the same dynamics in each location. The reset and the dynamics of the dependent variables may have to be determined by the independent variables.

For this last condition, I need to consider a norm $\|\cdot\|$ on $\mathbb{R}^k$, and the induced distance $d(\cdot,\cdot)$ between subsets of $\mathbb{R}^k$ defined as $d(A,B) = \inf\{\|a-b\| \mid a \in A \text{ and } b \in B\}$. From now on, two edges e and e' will be said to be *subsequent* if the target node of e is the source node of e'.

Definition 4.6 (Independent Dynamics Automata) *A hybrid automaton H is an* independent dynamics *automaton, or simply an* IDA*, if:*

1. *H is defined over a decidable theory over the reals;*
2. *For each pair of subsequent edges e and e' following holds: $d(\mathcal{R}(e), \mathcal{A}(e')) > 0$;*
3. *The vector Z of variables can be partitioned into two vectors X (independent variables) and Y (dependent variables) such that:*
   (a) *for each $e \in \mathcal{E}$ $Reset(e)[Z, Z']$ is of the form $(X' = X) \wedge \sigma(e)[Z, Y']$;*
   (b) *for each $v \in \mathcal{V}$ $Dyn(v)[Z, Z', T]$ is of the form $(X' = fi(X,T)) \wedge (Y = fd(v)(Z,T))$.*
4. *The set of values $p \in \mathbb{R}^k$ satisfying $Inv(v)[Z]$ is closed and bounded for each $v \in \mathcal{V}$.* □

Example 2 *Consider the hybrid automaton $H = (Z,\ Z',\ \mathcal{V},\ \mathcal{E},\ Inv,\ \mathcal{F},\ Act,\ Reset)$ such that: The dimension, k of the automata is 2; The discrete projection $\langle \mathcal{V}, \mathcal{E} \rangle$ is illustrated in Figure 1; The formulæ $Inv(v_1)[Z]$, $Inv(v_2)[Z]$ and $Inv(v_3)[Z]$ are equal to $Z_2 \geq (Z_1)^2 \ \wedge \ Z_2 \leq 100$; the remaining formulæ defining the automaton are:*

$$\begin{gathered}
Dyn(v_1)[Z,Z',T] \equiv Z_1' = 2(T)^3 + Z_1 \ \wedge \ Z_2' = -3T + Z_2 \\
Dyn(v_2)[Z,Z',T] \equiv Z_1' = -(T)^2 \ \wedge \ Z_2' = -3T + Z_2 \\
Dyn(v_3)[Z,Z',T] \equiv Z_1' = 2(T)^3 + Z_1 \ \wedge \ Z_2' = -3T + Z_2 \\
Reset(e_1)[Z,Z'] \equiv Z_1' \leq 8 \ \wedge \ Z_2' = Z_2 \\
Reset(e_2)[Z,Z'] \equiv Z_1' = Z_1 \ \wedge \ Z_2' = Z_2 \\
Reset(e_3)[Z,Z'] \equiv Z_1' = Z_1 \ \wedge \ Z_2' = Z_2 \\
Act(e_1)[Z] \equiv Z_2 > 5 \\
Act(e_2)[Z] \equiv Z_1 \ \wedge \ Z_2 \geq 0 \\
Act(e_3)[Z] \equiv (Z_1)^2 + (Z_2 - 5)^2 \leq 8
\end{gathered}$$

The automaton H is an IDA. □

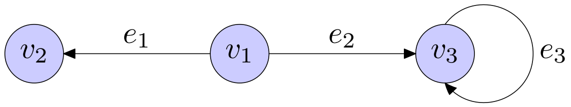

Figure 1: The discrete component of the Example 2.

It is not difficult to extend our class of automata by allowing different partitioning of the variables depending on the topology of the discrete structure. However, such an extension would involve many tedious technical details, which is preferably omitted in this short review.

Various questions about the power of IDAs can be found in [19]; I refer the interested readers to this paper and its

by solving a satisfiability problem.

sequels. But I will conclude this section with the following useful positive result.

Consider a class of automata over which an effective (terminating) algorithm can be devised to test reachability (see algorithm in [19]).

Definition 4.7 (∞IDA) *Let H be an IDA. H is said to be in the class ∞IDA if and only if for each edge $e = \langle v, v' \rangle$ of H it holds that f_v is continuous on $\mathbb{R}^k \times \mathbb{R}^+$ and for each $p \in \mathcal{I}(v)$ it holds that $\lim_{t \to +\infty} \| fi\,(p,t) \| = +\infty$.* □

Only some of the components of p are used in $fi\,(p,t)$, i.e., only the components of p corresponding to the independent variables.

Theorem 4.2 *Let H be an ∞IDA, ι and τ be two formulæ. The algorithm in [19] can be used to decide whether in H $Sat(\iota)$ can reach $Sat(\tau)$.* □

A few remarks are in order, especially in the context of classes of automata that my research group introduced in [19, 20, 21] in its effort to discover the right formalisms to take us beyond the linear hybrid systems. These, unlike many of the earlier generalizations, do not admit finite bisimulation quotients (a default approach of attack in this area). However, since for these automata, it is possible to define a translation from temporal formulae (a fragment of CTL) into first-order formulae over the reals, they permit effective model checking procedure—a property of enormous value in computational systems biology. In a following section, I will touch on some questions related to model checking and hybrid automata.

However, before leaving this section, I wish to highlight two questions in this area that are likely to grow in importance: (1) In systems biology, model checking methods for modular and hierarchical described hybrid models will need to be developed. However, it appears that decidability in a class of hybrid automata does not imply a similar decidability result for models constructed by the composition of members (Cartesian product) of the class. This apparent lack of a closure property is at a first glance rather bewildering, but could be intuitively explained by the difficulty in keeping track of how the modules within a system remain synchronized. (See [18].) (2) The second question deals with an obvious but neglected problem in systems biology: namely, from where do the hybrid automata come? Thus there is a need for algorithmic approaches to construct hybrid automata from experimental data. (See [17].)

4.3 Example

The utility of an IDA model may be demonstrated through a simple, yet biologically relevant example: namely, the *bacterial chemotaxis* [78, 12]. *Escherichia coli* has evolved an extremely effective strategy for responding to a chemical gradient in its environment, by detecting the concentration of ligands through a number of receptors, and then reacting to the input signal for driving its flagella motors to alter its path of motion. *E. coli* responds in one of two ways: either it "runs" – moves in a straight line by moving its flagella counterclockwise (CCW) (typically lasting 1000 ms), or it "tumbles" – randomly changes its heading by moving its flagella clockwise (CW) (typically lasting 100 s). The response is mediated through the molecular concentration of CheY in a phosphorylated form (Y_P variable in Figure 2), which in turn is determined by the bound ligands at the receptors that appear in several forms (LT variables in Figure 2). The ratio of $y = Y_P/Y_0$ (phosphorylated concentration of CheY to its concentration in the unphosphorylated form) determines a bias with an associated probability that flagella will exert a CW rotation; note that, in our example IDA model (Figure 2), I have simplified this situation by ignoring this stochastic effect by modelling it deterministically. Thus the most important output variable is the angular velocity w that takes discrete values +1 for CW and −1 for CCW. The more detailed pathway involves other molecules: CheB (either with phosphorylation or without, B_p and B_0), CheZ (Z), bound receptors (LT) and unbound receptors (T), while their continuous evolution is determined by a set of differential algebraic equations derived through kinetic mass action formulation.

This IDA model captures the essence of how an *E. coli* cell performs a biased random walk by transiently decreasing its tumbling frequency to move towards a region with greater ligand concentration. A second feature of this control is its sensitivity to concentration gradients and its observed dynamic range: rather than responding to absolute concentrations, the *E. coli* adapts quickly as it compares its environment during the immediate past to what existed a little earlier. Further, it does so over a wide range of input concentrations. □

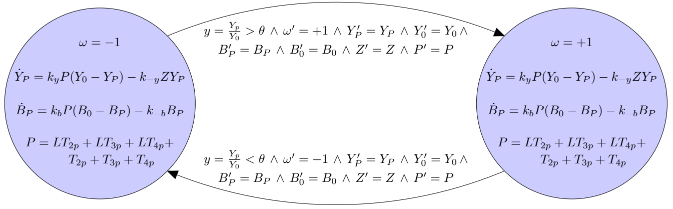
RUN [CCW]
TUMBLE [CW]
$\omega = -1$
$\dot{Y}_P = k_y P(Y_0 - Y_P) - k_{-y} Z Y_P$
$\dot{B}_P = k_b P(B_0 - B_P) - k_{-b} B_P$
$P = LT_{2p} + LT_{3p} + LT_{4p} + T_{2p} + T_{3p} + T_{4p}$
$\omega = +1$
$\dot{Y}_P = k_y P(Y_0 - Y_P) - k_{-y} Z Y_P$
$\dot{B}_P = k_b P(B_0 - B_P) - k_{-b} B_P$
$P = LT_{2p} + LT_{3p} + LT_{4p} + T_{2p} + T_{3p} + T_{4p}$
$y = \frac{Y_p}{Y_0} > \theta \wedge \omega' = +1 \wedge Y_P' = Y_P \wedge Y_0' = Y_0 \wedge B_P' = B_P \wedge B_0' = B_0 \wedge Z' = Z \wedge P' = P$
$y = \frac{Y_p}{Y_0} < \theta \wedge \omega' = -1 \wedge Y_P' = Y_P \wedge Y_0' = Y_0 \wedge B_P' = B_P \wedge B_0' = B_0 \wedge Z' = Z \wedge P' = P$

Figure 2: An IDA capturing the run-tumble mechanism of *E. coli.*

4.4 Model Checking in Systems Biology

While reachability analysis hints at the basic automated reasoning capability needed in systems biology, and rightfully occupies a pivotal role in reasoning about complex biological systems, it must also be abundantly clear to the reader that there exists a still bigger class of pertinent biological queries, statements and hypotheses that remain beyond the grasp of reachability. A semantically rich language for this purpose can be fashioned with the aid of a rather simple propositional temporal logic, which could in principle extend the techniques explored so far.

As examples of some straightforward biological property queries that occur naturally in the emerging field of computational systems biology, one may list the following: safety analysis, sensitivity analysis, robustness (both local and global) analysis, steady state or homeostasis properties, flux-balance analysis, etc.

For instance, one may consider the following model checking problem centering around verification of *safety properties*: given a biological system (e.g., encoded by a hybrid automaton H) and a safety property ϕ, one may wish to test whenever ϕ holds along all of H's trajectories. Since this is the case if and only if there is no reachable state in which ϕ does not hold, the verification of safety properties naturally reduces to the reachability problem. But, this situation represents an exception and not the rule, since evidently not all the properties of interest can be so easily translated to the reachability problem. Thus one must explore the situation for the most general classes of queries in the temporal logic of choice.

In a simpler setting, one may desire to represent biological systems as transition systems and apply classical model checking techniques (see e.g., [26]). As an example, consider the following example of robustness analysis of a model, representing the *purine metabolism pathway.* The main metabolite in purine biosynthesis is 5-phosphoribosyl-α-1-pyrophosphate (PRPP). A linear cascade of reactions converts PRPP into inosine monophosphate (IMP). IMP is the central branch point of the purine metabolism pathway. IMP is transformed into AMP and GMP. Guanosine, adenosine and their derivatives are recycled (unless used elsewhere) into hypoxanthine (HX) and xanthine (XA). XA is finally oxidized into uric acid (UA). In addition to these processes, there appear to be two “salvage” pathways that serve to maintain IMP levels and thus adenosine and guanosine levels as well. In these pathways, adenine phosphoribosyltransferase (APRT) and hypoxanthine-guanine phosphoribosyltransferase (HGPRT) combine with PRPP to form ribonucleotides. This pathway is believed to be extremely robust—not just to small perturbations, but significant and repeated perturbations to the input signal. In a branching time propositional temporal logic (e.g., CTL), one could formulate a query that would ask the model checker if the following formula is true (in a discretized simulation-preserving model):

```
Always (PRPP > 50 * PRPP1)
   implies
    steady_state()
    and Eventually(IMP1 > IMP2)
    and Eventually(HX < HX1)
    and Eventually(Always(IMP = IMP1))
    and Eventually(Always(HX = HX1))
```

However, when this particular query was examined with NYU's Simpathica/XSSYS tool, it was determined to be false, but pointed to certain incompleteness in the pathway representation of a commonly accepted model (shown in Figure 3). When the pathway representation was augmented with additional back-up reactions (already known in the literature), the robustness query turned out to be true, as determined by the same model checker.

Note, however, that, in general, hybrid automata have infinite state systems and the standard model checking techniques, which work on finite state models, fail. To solve this problem, many authors have suggested the use of equivalence reductions based on relations such as bisimulation. In particular, since bisimulation preserves branching-time temporal logics such as CTL and CTL*, whenever the bisimulation quotient of a system is finite, one could verify CTL and CTL* properties of the system by applying finite model checking techniques on its bisimulation quotient. Obviously, bisimulation has the advantage of preserving more expressive logics, but in many cases it produces infinite quotients. On the other hand, one may choose to use the simpler simulation relations *mutatis mutandis.* However, note that simulation preserves less expressive logics, while being more permissive as it can reduce a significantly larger class of automata to finite state models.

Furthermore, since on a single hybrid automaton one can consider both timed and untimed semantics, one can compute (bi)simulation on both of them. For these reasons, one must distinguish between the so called *timed-abstract simulations/bisimulations*, computed on the untimed semantics, and the *timed simulation/bisimulation*, evaluated on timed semantics.

While there are many versions of timed-semantics, with their individual quirks and idiosyncrasies, a particularly elegant formulation is given via TCTL.

Definition 4.8 TCTL[2] *It has the following syntactic*

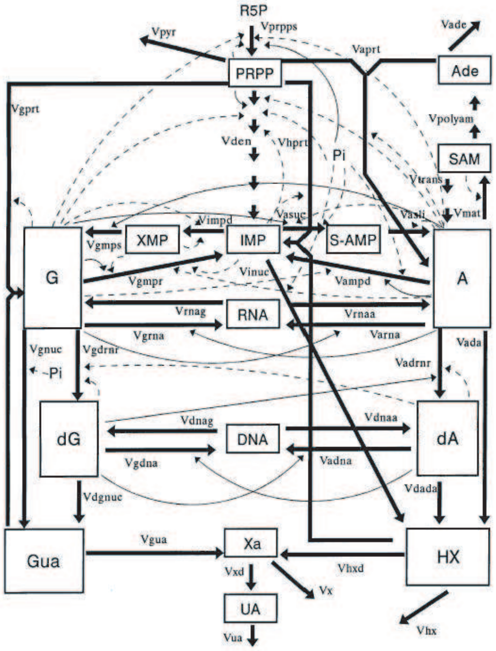

Figure 3: The pathway above depicts the classical (but incomplete) model of purine metabolism. Using Simpathica/XSSYS Systems Biology tools, one can automatically create a discretized model and perform model checking on the resulting automaton to show that the model fails to satisfy a particular temporal logic formula, encoding global robustness. This reasoning process and manipulations with model checking tools pointed to a more complete model that does satisfy the robustness formula.

structure:

$$\phi ::= p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \exists\mathcal{U}\phi_2 \mid \phi_1 \forall\mathcal{U}\phi_2 \mid z.\phi.$$

Its associated semantics is described below:

- **z.**: *The freeze quantification "z." binds the associated variable z to the current time. Thus the formula $z.\phi(z)$ holds at time t iff $\phi(t)$ does.*
- $\phi_1 \forall\mathcal{U}\phi_2$ *and* $\phi_1\ \exists\mathcal{U}\ \boldsymbol{\phi_2}$: *universal (on all paths) and existential (on at least one path) "until" operators. For $\phi_1\ \mathcal{U}\ \phi_2$ to be true on a path, ϕ_2 is required to be true somewhere along the path, and ϕ_1 is required to be true all along the path up to (but not necessarily at) that point.* □

Remark 4.3 *The basic notations are often extended by the following syntactic abbreviations [2].*

1. $p\ \exists\mathcal{U}_{\leq max}\ q \equiv p\ \exists\mathcal{U}\ (q \wedge z.(z \leq max))$ *and* $p\ \forall\mathcal{U}_{\leq max}\ q \equiv p\ \forall\mathcal{U}\ (q \wedge z.(z \leq max))$: *"subscripted"* Until *operators (max is the time-bound).*
2. *($\forall\mathcal{F}\ p \equiv$ true $\forall\mathcal{U}\ p$) and ($\exists\mathcal{F}\ p \equiv$ true $\exists\mathcal{U}\ p$: "eventuality" operators.*
3. *($\forall\mathcal{G}\ p \equiv \neg\exists\mathcal{F}\neg p$) and ($\exists\mathcal{G}\ p \equiv \neg\forall\mathcal{F}\neg p$): "invariance" operators.*

Definition 4.9 Single-Step Until Operator, $\triangleright$, [43]. *The formula $p \triangleright q$ holds if $p \vee q$ is true all along "one step" of the hybrid system and q is true at the end of the transition.* □

Definition 4.10 $T\mu$-Calculus Syntax: [43]. $\phi ::= X \mid p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \triangleright \phi_2 \mid z.\phi \mid \mu X.\phi$, *where μ is the* least-fixpoint *operator*[5]. *Thus,*

- Existential Until*:* $p\ \exists\mathcal{U}\ q = \mu X.(q \vee (p \triangleright X))$
- Universal Until[6]*:* $p \forall\mathcal{U} q = \neg(\neg q\ \exists\mathcal{U}\ (\neg p \wedge \neg q))$ □

In addition to using this logic to answer a particular instantiation of a biological model (e.g., with fixed values of system parameters), one may wish to engineer it appropriately to provide symbolic model checking facilities (e.g., the system parameters are treated as symbolic variables, and it is sought to determine various "equational relations" among these variables in order for the biological system to satisfy certain TCTL formulae). In other words, one seeks to devise algorithmic algebraic solutions to various kinds of queries (in TCTL) to examine interesting properties and invariants about the hybrid automata that model biochemical systems. As one would expect, the simplest and perhaps the most important question that one could ask about these systems is the symbolic state reachability problem: namely, can one reach a particular state from an initial state by following the dynamics of the hybrid automaton which may be described symbolically? For instance, a very relevant biological question would be to provide a symbolic description of the initial conditions (states) from which the biological system (modeled symbolic algebraically) can reach a desired state (say, apoptosis state for a cancer cell), or avoid certain unsafe states. In this sense, algebraic descriptions of this form in systems biology can be a potent tool, and need focused attention. We, as well as others, have made some progress by exploiting approximations, bounded reachability analysis, etc. or by suitably constraining the power of the family of hybrid automata studied [69, 67, 65, 20, 21, 19, 64, 66, 63, 16]. But much remains to be done!

[5] The *greatest-fixpoint* ν can be expressed as $\neg\mu X.(\neg\phi[X := \neg X])$.

[6] This translation is valid only when q is "finitely variable" over all premodels [43].

5 Historical and Bibliographic Notes

Since computational systems biology attempts to create a consilience of many different scientific fields, it must respond to the diverse styles, traditions, and history. What I provide below is a Cook's tour of that history without any pretense of completeness. One may rightfully argue that the present renaissance in computational systems biology owes great debts to the seminal paper of Watson and Crick [87, 86] and then the efforts that went into sequencing genomes of a diverse group of organisms, starting with the human genome project [15]. At present, it is possible to access and study, in great details, the genomes of a plethora of organisms: By the end of 2001, there existed assembled sequence data for *H. influenza*, *E. coli*, and thirty-nine other bacteria, *S. cerevisiae*, *C. elgans*, *D. melanogaster*, *A. thaliana* and a rough draft for *H. sapiens* [37, 49]. Subsequently, many more genomes were sequenced and became publicly available for further analysis. At present, this list includes: human (appearing on the covers of Science [49] and Nature [37]), chimpanzee, dog, chicken, rat and mouse (on covers of Nature [80, 50, 48, 29, 28], respectively), mosquito and fly (on covers of Science [73, 60], respectively), malaria parasite and slime mold (Nature [62, 55], respectively), etc. Computational approaches were developed to annotate genes and their regulators, as well as to compare orthologous genes across the species. These studies gave rise to the field of functional genomics and a qualitative version of systems biology [10]. Researchers were able to combine various "omics" (genomics to study genomes, transcriptomics to study RNA abundance, proteomics to study protein, regulomics, metabolomics, etc.) to piece together a picture of the tangled network of how these discretised elemental objects of molecular biology may be modelled to influence each other. The asymptotic topological structures of these networks and many over-abundant motifs they contained turned out to be exceptionally fascinating. They hinted not only at the elegance of the nature's solution at a static level, but also at the exquisite beauty of the evolutionary dynamics which sculpted these bio-chemical networks, it seemed appropriate that the field of systems biology should turn its attention to a quantitative analysis of these systems as various biomolecules interact within and across the cells—usually, categorized into three classes: regulatory, metabolic and signal-transduction pathways. The goal of this emerging field became a better understanding of the foundational design principles that persisted across the evolutionary scales. In this setting, systems biology discovered its roots in quantitative modelling of enzyme kinetics, a discipline that flourished between 1900 and 1970, but also in the simulations developed to study neurophysiology, the control and communication theories, and game theory. A particularly interesting approach to understand the tangles of biochemical reactions came from the formalisms of S-system and its extensions to XS-systems [84, 83, 8]. The complex networks of biochemical reactions could then be created out of elemental reactions such as: reversible and irreversible reactions, synthesis and degradation reaction (possibly, involving stoichiometric constraints) and enzyme or enzyme-coenzyme based reactions. Of course, such modules could then be abstracted at different levels of hierarchy and composed to create a hierarchical modular structure. Other approaches focusing on SBML [46] (Systems Biology Markup Language) and related, SBW (Systems Biology Workbench) also have become widely influential, because of the ease with which they could be used by working biologists.

The catalytic model of gene activation by transcriptional activators (either protein or micro-RNA factors) played a crucial role in genetic regulatory models and the resulting systems of ordinary differential reactions employed by systems biology. These models closely followed the classical Michaelis-Menten equations and its generalization in the form of Hill's equations. The word catalyst was coined in 1836 by the Swedish chemist Jöns Jakob Berzelius, who collected a number of examples of catalysis. In 1892, the British chemist Adrian Brown, had studied the rate of fermentation of sucrose in the presence of yeast to discover that the rate primarily depended on the invertase molecules that formed a complex with the sucrose, and had been influenced by the work of Cornelius O'Sullivan, Frederick Tompson (1890) and Charles Wurtz (1880). Along the way, the field has been invigorated by several seminal discoveries, as listed below: Fischer's "lock and key hypothesis" of 1894 (specificity of enzyme action, explained in terms of the precise fitting together of enzyme and substrate molecules); Victor Henri's 1903 research elucidating relationship between substrate and enzyme concentration; work of Leonor Michaelis and Maud Menten (1913), following earlier work by Archibald Hill, where kinetic consequences of an enzyme-substrate complex was used to give a precise formulation of Michelis-Menten Equation; finally, a more general formulation (of the Michaelis-Menten equation) by George Briggs and the British geneticist John Burdon Sanderson (J.B.S.) Haldane (in 1925).

The modern systems biology is often argued to have been an intellectual descendant of general systems theory, originally proposed by Ludwig von Bertalanffy. Some of the key steps in the metamorphosis of general systems theory into the present-day systems biology include the following: Work of the British neurophysiologists Alan Hodgkin and Andrew Huxley (1952) on a mathematical model of the action potential propagating along the axon of a neuronal cell; Denis Noble's (1960) computer model of a beating heart; development in the 60s and 70s of several approaches to study complex molecular systems, such as the Metabolic Control Analysis (MCA) and the Biochemical Systems Theory; many successes of molecular biology throughout the 80s, (although hindered by a skepticism toward theoretical biology that then promised more than it achieved); the birth of functional genomics in the 90s with accompanying large quantity of good quality data; Masaru Tomita and his group's (1997) quantitative model of the metabolism of a whole (hypothetical) cell, and creation of (around the year 2000) institutes and departments of Systems Biology in Seattle, Tokyo and Boston. Systems Biology may have finally found acceptance as a movement in its own right, spurred on by the completion of various genome projects, the large increase in data from the omics (e.g. genomics and proteomics) and the accompanying advances in high-throughput experiments and bioinformatics.

The field of model checking is likely to assume as important a role in systems biology as it is currently playing in many engineering and computational areas; this field is much younger but possesses an interesting history. Model checking, as a tool for systems verification, was proposed as an alternative to automatic proof-theoretic approaches, which are "in general quite tedious" and require "a good deal of ingenuity ... to organize the proof in a manageable fashion (Emerson and Clarke 1981 [27])." These highly practical and popular techniques are used to "mechanically determine if the system meets a specification expressed in propositional temporal logic." It has been shown, through ample examples, that model checking enjoys many advantages over its rivals: namely, it requires no proofs; it is fast (compared to other rigorous methods); it produces diagnostic counterexamples if the system's "real" behaviour does not match the believed (specified) behaviour; it is not hampered by partial specifications; and its underlying logics can easily express main properties of interests in many domains of application.

In parallel to the work of Clarke et al. and Pnueli et al., there have appeared several techniques that share a common philosophy in their approach to the problem of reasoning about large complex systems: namely, the independent work of Tadeo Murata and Kurt Jensen on Petrinet tools (late 70s); Gregor Bochmann's work on Protocol Verification (1978) as well as the work of Holzman also on Protocol Verification, (1978-79).

The interest in temporal logic as a language for specifying computational protocols and programs date back to the mid seventies: e.g., the work of Burstall (1974), of Kroeger (1977), and of Pnueli (1977), all proposing temporal logic for reasoning about computer programs. However, Amir Pnueli first used temporal logic for reasoning about concurrency, as he proved program properties from a set of axioms that described the behavior of the individual statements. The method was extended to sequential circuits by others (e.g., Bochmann in 1982 and Owicki and Malachi in 1981). In 1981, Edmund Clarke and E. Allen Emerson wrote an influencial paper (" Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic." Logics of Programs Workshop, Yorktown Heights, New York, May 1981, LNCS 131) showing that their temporal logic model checking algorithms allowed this type of reasoning to be automated. The success of this approach can be attributed to the simple fact that "Checking that a single model satisfies a formula is much easier than proving the validity of a formula for all models." Their algorithm was practical (with polynomial time complexity). In mid 80's, the present author and Edmund Clarke showed its use for Hardware Verification (and discovered a thorny hardware bug in a published circuit designed by Charles Seitz to implement a FIFO Queue in hardware).

Another innovation, with a high practical payoff, in some of the early model checkers (e.g., CMU's Extended Model Checker (EMC)) was the facilities for giving counterexamples for universal CTL (computational tree logic) properties that were false or witnesses for existential properties that were true. This was added by Michael Browne in 1984 to an enhanced implementation, called the MCB model checker. Further enhancements include: Automata Theoretic Techniques of Aggarawal, Kurshan and Sabnani (1983), also explored in David Dill's Thesis (1987), and by Har'El et al. (1990); and two major breakthroughs in the field in the form of "symbolic model checking" techniques due to two teams of researchers, Coudert, Berthet, and Madre (1989) and Burch, Clarke, McMillan, Dill, and Hwang (1990) and the "partial order reduction" techniques due to Valmari (1990), Godefroid (1990) and Peled (1994).

However, the field is far from mature, as it still needs to deal with many problems that arise naturally in very complex systems as those appearing in biology. There is a strong impetus to explore non-traditional techniques: namely, techniques based on (i) compositional reasoning, (ii) abstraction, (iii) symmetry reduction, and (iv) induction and parameterized verification. New model checking techniques have pushed the field in novel directions: timed and hybrid automata (a topic likely to be very important to systems biology), bounded model checking, localization reduction, compositional model checking and learning, and infinite state systems. The field continues to grow in momentum, as many of its early practitioners begin to be recognized for the field's impact [7]

A rigorous attempt to create tools for systems biology would most likely depend on various algebraic techniques: both from commutative as well as differential algebra. These techniques must be effective (i.e., algorithmic) and efficient (i.e, have low computational complexity) in order for the tools to be practical, rigorous and scalable. Interestingly, the history of algorithms and algebra is somewhat intertwined: Abū 'Abd Allāh Muhaammad ibn Mūsā al-Khwārizmī (780-850 AD), a mathematician in the court of Caliph Harun Al Rasid of Abassid Dynasty, is credited as the father of both of these subjects. Two of his books, Al-Kitab al-Mukhtasar fi-hisab al-Jabr al-Muqabalah (Algebra) and Kitab al-Jam'a wal-Tafreeq bil-Hisab al-Hindi (Algorithm), established the foundations for these two fields. The later of the two, translated into Latin in the twelfth century, as Algoritmi de numero Indorum, and incorporating many of the ideas in Aryabhatta's Siddhanta (also translated by al-Khwārizmī into Arabic as SindHind) created a revolutionary amalgamation of Indian and Greek mathematics

Soon after that, the field of algebra saw a profusion of breakthroughs; a short list of some of the important milestones in the history of algebra would include: circa 850: Persian mathematician al-Mahani conceived the idea of reducing geometrical problems such as duplicating the cube to problems in algebra; circa 850: Indian mathematician Mahavira solved various quadratic, cubic, quartic, quintic and higher-order equations, as well as indeterminate quadratic, cubic and higher-order equations; circa 990: Persian Abu Bakr al-Karaji, further develops algebra by replacing geometrical operations of algebra with modern arithmetical operations, and defining the monomials; circa 1050: Chinese mathematician Jia Xian discovered numerical solutions of polynomial equations; circa1072: Persian mathematician Omar Khayyam developed algebraic geometry and, gave a complete classification of cubic equations; circa 1114: Indian mathematician Bhaskara, in his Bijaganita (Algebra), solved various cubic, quartic and higher-order polynomial equations, as well as the general quadratic indeterminant equation; circa 1202: Leonardo Fibonacci of Pisa through his work Liber Abaci introduced the subject to Europe; circa 1300: Chinese mathematician Zhu Shijie created polynomial algebra, and solved simultaneous equations etc.; circa 1400: Indian mathematician Madhava of Sangamagramma gave iterative methods for approximate solution of non-linear equations; circa 1545: Girolamo Cardano published Ars magna, which elucidated Fontana's solution to the general quartic equation; circa 1591: Francois Viete developed improved symbolic notation In artem analyticam isagoge; circa 1682: Gottfried Leibniz developed his notion of symbolic manipulation with formal rules, called characteristica generalis; circa 1750: Gabriel Cramer stated Cramer's rule and studied algebraic curves, matrices and determinants; circa 1824: Niels Henrik Abel proved that the general quintic equation is insoluble by radicals; circa 1832: Evariste Galois developed Galois theory; circa 1950: Tarski gave a decision method for elementary algebra and geometry, for which the first elementary recursive method was found by Collins using the technique of Cylindrical Algebraic Decomposition (CAD), with a better complexity (doubly exponential.) The field of algorithmic algebra that combined computational methods with algebraic manipulation to analyze systems of algebraic equations, inequations and inequalities flourished with the development of many beautiful algorithms: Buchberger's algorithm for Gröbnr bases, Ritt's characteristic sets, Tarski's algorithm for semi-algebraic geometry, and the methods based on resultants. As we described, many of these techniques have begun to influence the development of effective techniques to decide various properties of hybrid automata, and thus simpler but immensely informative models of biochemical process. The emergence of the field of "Algebraic Biology" is a first step in combining ideas from algebra, algorithms and automata to provide tools to understand biology better. This review paper has been inspired by this consilience, which has begun to attract practitioners from many different fields and with many different backgrounds. It is hoped that these pioneers will build upon the foundations that is being laid now and create an enduring structure.

6 Challenges of systems Biology

It may be considered premature to attempt listing what problems would dominate the thinking in this emerging field for the next few years; nonetheless, we suggest a handful:

1. **Problem 1: What are the most important hybrid systems models for biology?**

   These models should have many desirable properties: namely, high fidelity, expressivity, and decidability (and computational efficiency).

2. **Problem 2: How do genotypes determine phenotypes?**

[7] For instance, in recognition of model checking's impact on both theoretical and practical computer science, Amir Pnueli, Ed Clarke, Allen Emerson, Joe Sifakis, Ken McMillan, Randy Bryant, Robert Kurshan, David Harel, etc. have been awarded some of the highest awards in computer science (Turing and Kanellakis awards, for instance).

We can take the view that the model (e.g., structure of the hybrid systems and their parameters, which may be inferred directly from the linguistic patterns in the genomes) encode the genotypes, where as the temporal logic formulæ satisfied by the model encode the phenotypes. One may then ask: How does biology relate them? Are there engineering principles governing them? What properties are important? Are some parameters more "flexible," "facilitated" or "robust" than others? Are there symmetries? Scaling laws? Laws governing compositionality? Laws governing modularity and hierarchy? How does evolution control them? What structural changes can be accomplished by nature? What selection forces act on them? What invariants are important to biology? Are there utility functions that are being optimized? What are they? A similar, but a more detailed viewpoint, has been expounded by Harel and Keller[53].

3. **Problem 3: How can our understanding of systems biology lead to designing useful artificial biological systems?**

   Can we create ab initio a complete artificial organism? Can we perturb the properties of an existing cell by introducing synthesized biological circuits? These questions will see application of analytical techniques of systems biology to delineate the engineering design space.

4. **Problem 4: How do we build models? What measurements are important in this process?**

   Causal models vs. phenomenological models: Will it be possible now to create a causal model for any biological phenomenon, given our cognitive incompleteness of biological processes? Can we achieve a useful, albeit approximate, understanding of biology from simpler phenomenological models that may ignore details of many of the structural properties of models?

5. **Problem 5: How can we apply our understanding of systems biology to solve important biomedical problem?**

   The application areas would involve creation of new tools for diagnostics and prognostics, drug discovery, vaccine design, etc. Systems biology models, model checking and model-based supervisory controller design could become the basis for rational drug design in the future.

7 Closing Thoughts

Before closing, a reflection on the nature of scientific discovery is in order. Lest some may mistakenly conclude that I have argued parochially in favour of theories over experiments (equivalently, Newton over Hooke), I conclude this review with the following beautiful quote from Hooke:

> "So many are the links, upon which the true Philosophy depends, of which, if any can be loose, or weak, the whole chain is in danger of being dissolved; it is to begin with the Hands and Eyes, and to proceed on through the Memory, to be continued by the Reason; nor is it to stop there, but to come about to the Hands and Eyes again, and so, by a continuall passage round from one Faculty to another, it is to be maintained in life and strength."

It is hoped that in the near future we will see the emergence of the field of systems biology, serving as a strong link between biological principles and observations— maintained in life and strength, as Hooke had prophesied!

8 A Few Words of Gratitude

The ideas presented here owe enormously to several, but most prominently to: Carla Piazza, Alberto Casagrande and Venkat Mysore. In addition, I wish to thank all who have wittingly or otherwise contributed to my (still unfinished) journey into systems biology: Ed Clarke, Merrick Furst, Bob Tarjan, Dana Scott, Mike Foster, Brad White, Tom Anantharaman, Randy Bryant, Jeannette Wing, Prasad Sistla, Allen Emerson, David Dill, Bob Kurshan, Amir Pnueli, Rajeev Alur, Jack Schwartz, Martin Davis, Gerardo Lafferriere, Anne Dinning, Lou Salkind, Dayton Clark, David Shaw, Pasquale Cainiello, Chee Yap, Giovanni Gallo, Pina Carra, William Sit, Tom Dube, Paul Pedersen, Bruno Buchberger, Franz Winkler, Bernd Sturmfels, Volker Weispfenning, Teo Mora, Michael Singer, Ricky Pollack, Marie Francois-Roy, Saugata Basu, Mohsen Jafari, Marco Antoniotti, Shankar Sastry, Toto Paxia, Roger Brockett, Sanjoy Mitter, Pravin Variya, Pradeep Khosla, Dan Koditschek, John Canny, Bruce Donald, Eduardo Sontag, Rohit Parikh, Laxmi Parida, Raj Reddy, Izzy Edelman, Misha Gromov, Ale Carbone, Mike Wigler, Jim Watson, Joey Zhou, Raoul Daruwala, David Schwartz, Mike Waterman, Dick Karp, Ron Shamir, Gene Myers, J. Craig Venter, Gilad Lerman, Joe McQuown, Jim Glimm, Raphy Coiffman, Charles Cantor, Lee Hood, Charles Delisi, Shree Kumar, Ashish Tiwari, Vijay Kumar, Pat Lincoln, Jim Collins, Drew Endy, George Pappas, Claire Tomlin, Pablo Parillo, Calin Belta, Nadia Ugel, Paolo Barbano, Marina Spivak, Jiawu Feng, Yuri Lazebnik, Jack Lin, Seongho Ryu, Jane Hubbard, Fang Cheng, Joan Brugge, David Harel, Marc Kirshner, Partha Mitra, John Doyle, Richard Murray, Mike Savageau, Naren Ramakrishnan, Debo Das, Bahram Parvin, Jeff Trent, R.K. Shyama Sundar, P.S. Thiagarajan, Somenath Biswas, Manindra Agarwal, Alberto Policriti, Raffaella Gentilini, Iuliana Ionita, Samantha Kleinberg, Kevin Casey, Giuseppe Narzisi, Giuseppe Nicosia, Antonina Mitrofanova, Mike Teitell, Andrew Sundstrom, Alex Lim, Pierre Franquin, Jason Reed, Jim Gimzewski, Kalim Mir and Patrick Soon-Shiong. This paper is dedicated in memory of a fellow traveler and close friend, Pina Carra, who is dearly missed.

References

[1] R. Alur, C. Belta, V. Kumar, M. Mintz, G. J. Pappas, H. Rubin, and J. Schug. Modeling and Analyzing Biomolecular Networks. *Computing in Science and Engineering*, 4(1):20–31, January 2002.

[2] R. Alur, C. Courcoubetis, and D. Dill. Model-Checking for Real-Time Systems. In *International Symposium on Logic in Computer Science*, 5, pages 414–425. IEEE Computer Press, 1990.

[3] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The Algorithmic Analysis of Hybrid Systems. *Theoretical Computer Science*, 138(1):3–34, Febrary 1995.

[4] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *LNCS*, pages 209–229, London, UK, 1993. Springer-Verlag.

[5] R. Alur, T. Dang, and F. Ivancic. Reachability analysis of hybrid systems via predicate abstraction. In C. Tomlin and M. R. Greenstreet, editors, *Proceedings of the Fifth Workshop on Hybrid Systems: Computation and Control (HSCC 2002)*, volume 2289 of *LNCS*, pages 35–48, Stanford, CA, USA, March 25-27 2002. Springer-Verlag.

[6] K. Amonlirdviman, R. Ghosh, J.D. Axelrod, and C.J. Tomlin. A Hybrid Systems Approach to Modeling and Analyzing Planar Cell Polarity. In *Proceedings of the 3rd International Conference on Systems Biology*, 2002.

[7] M. Antoniotti, C. Piazza, A. Policriti, M. Simeoni, and B. Mishra. Taming the complexity of biochemical models through bisimulation and collapsing: Theory and practice. *Theoretical Computer Science*, 325(1):45–67, 2004.

[8] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. XS-systems: eXtended S-Systems and Algebraic Differential Automata for Modeling Cellular Behavior. In *Proceedigs of the International Confernce on High Performance Computing, HiPC 2002*, Bangalore, India, December 2002.

[9] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Reasoning about Biochemical Processes. *Cell Biochemistry and Biophysics*, 38:271–286, 2003.

[10] B. Mishra et al. A Sense of Life: Computational and Experimental Investigations with Models of Biochemical and Evolutionary Processes. *OMICS*, 7:253–268, 2003.

[11] G. Batt, H. de Jong, J. Geiselmann, and M. Page. Qualitative Analysis of Genetic Regulatory Networks: A Model-Checking Approach. In *B. Bredeweg, P. Salles (eds.), Working Notes of Seventeenth International Workshop on Qualitative Reasoning, QR-03*, pages 31–38, 2003.

[12] H.C. Berg. Motile behavior of bacteria. *Physics Today*, 53(1):24–29, 2000.

[13] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1997.

[14] T. Brihaye, C. Michaux, C. Rivière, and C. Troestler. On o-minimal hybrid systems. In R. Alur and G. J. Pappas, editors, *Proceedings of Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *LNCS*, pages 219–233. Springer-Verlag, 2004.

[15] C.R. Cantor. Orchestrating the Human Genome Project. *Science*, 248:49–51, 1990.

[16] A. Casagrande. *Hybrid Systems: A First-Order Approach to Verification and Approximation Techniques*. PhD thesis, Department of Mathematics and Computer Science, University of Udine, 2006. Advisers - A. Policriti and T. Villa.

[17] A. Casagrande, K. Casey, R. Falchi, C. Piazza, B. Ruperti, G. Vizzotto, and B. Mishra. Translating Time-course Gene Expression Profiles into Semi-Algebraic Hybrid Automata via Dimensionality Reduction. In *Algebraic Biology (AB'07)*, LNCS. Springer, 2007.

[18] A. Casagrande, P. Corvaja, C. Piazza, and B. Mishra. Composing Semi-algebraic O-minimal Automata. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control (HSCC'07)*, volume 4416 of *LNCS*, pages 668–671. Springer, 2007.

[19] A. Casagrande, V. Mysore, C. Piazza, and B. Mishra. Independent Dynamics Hybrid Automata in Systems Biology. In *Proceedings of the First International Conference on Algebraic Biology (AB'05)*, pages 61–73, Tokyo, Japan, November 2005. Universal Academy Press, Inc.

[20] A. Casagrande, C. Piazza, and B. Mishra. Semi-Algebraic Constant Reset Hybrid Automata - SACoRe. In *Proceedings of the 44rd Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, pages 678–683, Seville, Spain, December 2005. IEEE Computer Society Press.

[21] A. Casagrande, C. Piazza, A. Policriti, and B. Mishra. Inclusion Dynamics Hybrid Automata. Submitted to journal, 2007.

[22] M. Cascante, L.G. Boros, B. Comin-Anduix, P. de Atauri, J.J. Centelles, and P.W.-N. Lee. Metabolic Control Analysis in Drug Discovery and Design. *Nature Biotechnology*, 20:243–249, 2002.

[23] N. Chabrier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and Querying Biochemical Interaction Networks. *Theoretical Computer Science*, 325(1):25–44, September 2004.

[24] N. Chabrier and F. Fages. Symbolic Model Checking of Biochemical Networks. In *Proceedings of the First International Workshop on Computational Methods in Systems Biology*, pages 149 – 162, 2003.

[25] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic Verification of Finite-State Concurrent Systems using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[26] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 1999.

[27] E.M. Clarke and E.A. Emerson. Synthesis of Synchronization Skeletons for Branching Time Temporal Logic. In *Logic of Programs: Workshop*, volume 131 of *LNCS*. Springer, 1981.

[28] Mouse Genome Sequencing Consortium. Initial Sequencing and Comparative Analysis of the Mouse Genome. *Nature*, 420:520–562, 2002.

[29] Rat Genome Sequencing Project Consortium. Genome Sequence of the Brown Norway Rat Yields Insights into Mammalian Evolution. *Nature*, 428:493–521, 2004.

[30] A. Cornish-Bowden. *Fundamentals of Enzyme Kinetics (3rd edn.).* Portland Press, London, 2004.

[31] A. Cornish-Bowden and M. L. Cardenas. Metabolic Analysis in Drug Design. *C. R. Biologies*, 326:509–515, 2003.

[32] A. Cornish-Bowden and M.L. Cardenas. Systems Biology may Work When We Learn to Understand the Parts in Terms of the Whole. *Biochemical Society Transactions*, 33(3), 2005.

[33] H. de Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology*, 9(1):69 – 105, 2002.

[34] H. de Jong. Modeling and Simulation of Genetic Regulatory Networks. *Lectures Notes in Control and Information Sciences*, 294:111–118, 2003.

[35] S. Drulhe, G. Ferrari-Trecate, H. de Jong, and A. Viari. Reconstruction of Switching Thresholds in Piecewise-Affine Models of Genetic Regulatory Networks. In *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006. Proceedings*, volume 3927 of *Lecture Notes in Computer Science*, pages 184–199, 2006.

[36] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway Logic: Symbolic Analysis of Biological Signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, January 2002.

[37] E.S. Lander et al. Initial Sequencing and Analysis of the Human Genome. *Nature*, 409:860–921, 2001.

[38] R. Ghosh, A. Tiwari, and C. Tomlin. Automated Symbolic Reachability Analysis; with Application to Delta-Notch Signaling Automata. In O. Maler and A. Pnueli, editors, *Int.l Workshop on Hybrid Systems: Computation and Control (HSCC'03)*, volume 2623 of *LNCS*, pages 233–248. Springer-Verlag, 2003.

[39] R. Ghosh and C. Tomlin. Lateral Inhibition through Delta-Notch Signaling: A Piecewise Affine Hybrid Model. In M. D. Di Benedetto and A. L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *LNCS*, pages 232–246. Springer-Verlag, 2001.

[40] R. Ghosh and C. Tomlin. An Algorithm for Reachability Computations on Hybrid Automata Models of Protein Signaling Networks. In *Proceedings of the 44rd Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, pages 2256–2261, Seville, Spain, December 2005. IEEE Computer Society Press.

[41] T. A. Henzinger and P. W. Kopke. State equivalences for rectangular hybrid automata. In U. Montanari and V. Sassone, editors, *Proceedings of International Conference on Concurrency Theory (Concur'96)*, volume 1119 of *LNCS*, pages 530–545. Springer-Verlag, 1996.

[42] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing (STOC '95)*, pages 373–382, New York, NY, USA, 29 May–1 June 1995. ACM Press.

[43] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic Model Checking for Real-time Systems. In *7th Annual IEEE Symposium on Logic in Computer Science*, pages 394–406. IEEE, IEEE Computer Society Press, June 1992.

[44] T.A. Henzinger. The Theory of Hybrid Automata. In *Proc. of IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 278–292. IEEE Press, 1996.

[45] J. Hu, W.-C. Wu, and S. Sastry. Modeling Subtilin Production in Bacillus subtilis using Stochastic Hybrid Systems. In R. Alur and G.J. Pappas, editors, *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004, Proceedings*, volume 2993 of *Lecture Notes in Computer Science*, pages 417–431. Springer, 2004.

[46] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J. Doyle, and H. Kitano. The ERATO Systems Biology Workbench: Enabling Interaction and Exchange between Software Tools for Computational Biology. In *In The Proceedings of the Pacific Symposium on Biocomputing*, 2002.

[47] T. Ideker and D. Lauffenburger. Building with a Scaffold: Emerging Strategies for High- to Low-Level Cellular Modeling. *Trends in Biotechnology*, 21(6):255–62, Jun 2003.

[48] International Chicken Genome Sequencing Consortium. Sequence and Comparative Analysis of the Chicken Genome Provide Unique Perspectives on Vertebrate Evolution. *Nature*, 432:695–716, 2004.

[49] J. Craig Venter et al. The Sequence of the Human Genome. *Science*, 291:1304–1351, 2001.

[50] K. Lindblad-Toh et al. Genome Sequence, Comparative Analysis and Haplotype Structure of the Domestic Dog. *Nature*, 438:803–819, 2005.

[51] K. Kappler, R. Edwards, and L. Glass. Dynamics in High Dimensional Model Gene Networks. *Signal Processing*, 83:789–798, 2002.

[52] J.P. Keener and J. Sneyd. *Mathematical Physiology.* Springer, New York, 1998.

[53] E Fox Keller and D. Harel. Beyond the Gene. *PLoS ONE*, 2, 2007.

[54] P. W. Kopke. *The Theory of Rectangular Hybrid Automata.* PhD thesis, Faculty of the Graduate School, Cornell University, 1996. Adviser - Thomas A. Henzinger.

[55] L. Eichinger et al. The Genome of the Social Amoeba Dictyostelium discoideum. *Nature*, 435:43–57, 2005.

[56] G. Lafferriere, G. J. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13:1–21, 2000.

[57] P. Lincoln and A. Tiwari. Symbolic Systems Biology: Hybrid Modeling and Analysis of Biological Networks. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 660–672. Springer, March 2004.

[58] J. Lygeros, K. H. Johansson, S. N. Simić, J. Zhang, and S. Sastry. Continuity and invariance in hybrid automata. In *Proceedings of the Fortieth IEEE Conference on Decision and Control (CDC '01)*, Orlando, Florida, USA, December 2001. IEEE Computer Society Press.

[59] O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems. In J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, volume 600, pages 447–484. Springer-Verlag, 3–7 June 1991.

[60] M.D. Adams et al. The Genome Sequence of Drosophila melanogaster. *Science*, 287:2185–2195, 2000.

[61] B. Mishra. A Symbolic Approach to Modeling Cellular Behavior. In S. Sahni, V. K. Prasanna, and U. Shukla, editors, *In Proceedings of the International Conference on High Performance Computing*, volume 2552 of *LNCS*, pages 725–732. Springer-Verlag, 2002.

[62] M.J. Gardner et al. Genome Sequence of the Human Malaria Parasite Plasmodium falciparum. *Nature*, 419:498–511, 2002.

[63] V. Mysore. *Algorithmic Algebraic Model Checking: Hybrid Automata & Systems Biology.* PhD thesis, Department of Computer Science, New York University, 2006. Advisors - B. Mishra.

[64] V. Mysore, A. Casagrande, C. Piazza, and B. Mishra. Tolque – A Tool for Algorithmic Algebraic Model Checking. In *The Ninth International Workshop on Hybrid Systems Computation & Control (HSCC06) Poster Session*, March 2006.

[65] V. Mysore and B. Mishra. Algorithmic Algebraic Model Checking III: Approximate Methods. In *Infinity 2005 – The 7th International Workshop on Verification of Infinite-State Systems*, volume 149(1) of *Electronic Notes in Theoretical Computer Science*, pages 61–77, February 2006.

[66] V. Mysore and B. Mishra. Algorithmic Algebraic Model Checking IV: Metabolic Networks. In *Algebraic Biology 2007*, volume NCS 4545, pages 170–184, July 2007.

[67] V. Mysore, C. Piazza, and B. Mishra. Algorithmic Algebraic Model Checking II: Decidability of Semi-Algebraic Model Checking and its Applications to Systems Biology. In Doron A. Peled and Yih-Kuen Tsay, editors, *Automated Technology for Verification and Analysis: Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005. Proceedings*, volume 3707 of *LNCS*, pages 217–233. Springer-Verlag, Oct 2005.

[68] M. Nagasaki, A. Doi, H. Matsuno, and S. Miyano. Genomic Object Net: A Platform for Modeling and Simulating Biopathways. *Applied Bioinformatics*, 2003.

[69] C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, and B. Mishra. Algorithmic Algebraic Model Checking I: The Case of Biochemical Systems and their Reachability Analysis. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005. Proceedings*, volume 3576 of *LNCS*, pages 5–19. Springer-Verlag, 2005.

[70] U. Platzer and H.-P. Meinzer. Simulation of Genetic Networks in Multicellular Context. In *D , Kim J. und Martinez T Polani (ed.), Fifth German workshop on Artificial Life: Abstracting and Synthesizing the Principles of Living Systems*, pages 43–51. Akad. Verl.-Ges., 2002.

[71] A. Pnueli. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science*, 13:45–60, 1981.

[72] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In D. L. Dill, editor, *Proceedings of International Conference on Computer Aided Verification (CAV'94)*, volume 818 of *LNCS*, pages 95–104. Springer-Verlag, 1994.

[73] R.A. Holt et al. The Genome Sequence of the Malaria Mosquito Anopheles gambiae. *Science*, 298:129–149, 2002.

[74] K. Sachs, D. Gifford, T. Jaakkola, P. Sorger, and D.A. Lauffenburger. Bayesian Network Approach to Cell Signaling Pathway Modeling. In *Sci. STKE*, 2002.

[75] I. Shmulevich, I. Gluhovsky, R.F. Hashimoto, E.R. Dougherty, and W. Zhang. Steady-State Analysis of Genetic Regulatory Networks Modelled by Probabilistic Boolean Networks. *Comparative and Functional Genomics*, 4:601–608, 2003.

[76] M. Sirava, T. Schafer, M. Eiglsperger, M. Kaufmann, O. Kohlbacher, E. Bornberg-Bauer, and H.P. Lenhof. BioMiner - Modeling, Analyzing, and Visualizing Biochemical Pathways and Networks. *Bioinformatics*, 18(2), 2003.

[77] B.M. Slepchenko, J.C. Schaff, I. Macara, and L.M. Loew. Quantitative Cell Biology with the Virtual Cell. *Trends in Cell Biology*, 13(11):570–576, November 2003.

[78] P. A. Spiro, J. S. Parkinson, and H. G. Othmer. A model of excitation and adaptation in bacterial chemotaxis. *Proc. Natl. Acad. Sci. USA*, 94:72637268, July 1997.

[79] K. Takahashi, N. Ishikawa, Y. Sadamoto, H. Sasamoto, S. Ohta, A. Shiozawa, F. Miyoshi, Y. Nakayama, and M. Tomita. E-CELL2: Multi-platform E-CELL Simulation System. *Bioinformatics*, 19(13):1727–1729, 2003.

[80] The Chimpanzee Sequencing and Analysis Consortium. Initial Sequence of the Chimpanzee Genome and Comparison with the Human Genome. *Nature*, 437:69–87, 2005.

[81] A. Tiwari and G. Khanna. Series of Abstraction for Hybrid Automata. In C. Tomlin and M. R. Greenstreet, editors, *Proceedings of the Fifth Workshop on Hybrid Systems: Computation and Control (HSCC 2002)*, volume 2289 of *LNCS*, pages 465–478, Stanford, CA, USA, March 25-27 2002. Springer-Verlag.

[82] A. M. Turing. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237:37–72, 1952.

[83] E. O. Voit. *Computational Analysis of Biochemical Systems. A Pratical Guide for Biochemists and Molecular Biologists.* Cambridge University Press, 2000.

[84] E. O. Voit and M. Savageau. Equivalence between S-systems and Volterra systems. *Mathematical Biosciences*, 78:47–55, 1986.

[85] G. von Dassow, E. Meir, E.M. Munro, and G.M. Odell. The Segment Polarity Network is a Robust Developmental Module. *Nature*, 406:188–92, 2000.

[86] J.D. Watson and F.H.C. Crick. Genetical Implications of the Structure of Deoxyribonucleic Acid. *Nature*, 171:964–967, 1953.

[87] J.D. Watson and F.H.C. Crick. A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738, 1953.

[88] L. Wolpert. Positional Information and the Spatial Pattern of Cellular Differentiation. *J. Theor. Biol.*, 25:1–47, 1969.