Pacific Symposium on Biocomputing, January 3-7, 2002, p400-412

# PATHWAY LOGIC: SYMBOLIC ANALYSIS OF BIOLOGICAL SIGNALING

STEVEN EKER, MERRILL KNAPP, KEITH LADEROUTE, PATRICK LINCOLN, JOSE MESEGUER, AND KEMAL SONMEZ

*SRI International 333 Ravenswood Ave, Menlo Park, CA, 94025*
*E-mail: lincoln@csl.sri.com*

The genomic sequencing of hundreds of organisms including *homo sapiens*, and the exponential growth in gene expression and proteomic data for many species has revolutionized research in biology. However, the computational analysis of these burgeoning datasets has been hampered by the sparse successes in combinations of data sources, representations, and algorithms. Here we propose the application of symbolic toolsets from the formal methods community to problems of biological interest, particularly signaling pathways, and more specifically mammalian mitogenic and stress responsive pathways. The results of formal symbolic analysis with extremely efficient representations of biological networks provide insights with potential biological impact. In particular, novel hypotheses may be generated which could lead to wet lab validation of new signaling possibilities. We demonstrate the graphic representation of the results of formal analysis of pathways, including navigational abilities, and describe the logical underpinnings of the approach. In summary, we propose and provide an initial description of an algebra and logic of signaling pathways and biologically plausible abstractions that provide the foundation for the application of high-powered tools such as model checkers to problems of biological interest.

## 1 Introduction

*Biological Signaling Pathways.* The tremendous growth of genomic sequence information combined with technological advances in the analysis of global gene expression has revolutionized research in biology and biomedicine[1]. However, the vast amounts of experimental data and associated analyses now being produced have created an urgent need for new ways of integrating this information into theoretical models of cellular processes for guiding hypothesis creation and testing. Investigation of mammalian signaling processes, the molecular pathways by which cells detect, convert, and internally transmit information from their environment to intracellular targets such as the genome, would greatly benefit from the availability of such predictive models. Although signaling pathways are complex, fundamental concepts have emerged from contemporary research indicating that they are amenable to analysis by computational methods. For example, most signaling pathways involve the hierarchical assembly in space and time of multi-protein complexes or modules that regulate the flow of information according to logical rules[2,3]. Moreover, these pathways are embedded in networks having stimulatory,

inhibitory, cooperative, and other connections to ensure that a signal will be interpreted appropriately in a particular cell or tissue[4,5].

*Modeling Cellular Signaling Networks.* Various models for the computational analysis of cellular signaling networks have been proposed involving approaches that incorporate rate and/or concentration information[6,7]. However, these critical approaches are currently limited by the great difficulty of obtaining true intracellular rate or concentration information. Moreover, they could be limited by the potentially stochastic features of cellular scale populations of signaling molecules[8]. Because of these problems, we have chosen to focus exclusively on an abstract level involving the logic of signal. Previous work at a similar level of abstraction includes EcoCyc, the pathway/genome and metabolic reaction database for *E. coli*[9,10]. Initial steps allowing simulation (Section 2.1) of biological pathways include work animating the EcoCyc database[11], and the use of π-calculus to represent and forward-simulate a small signaling pathway[12]. Here we describe an approach to the development of logical models based on the application of formal methods tools to mammalian signaling pathways[13,14]. Specifically, we describe the application of rewriting logic to the symbolic representation of a major receptor-mediated pathway in mammalian cells: receptor tyrosine kinase (RTK) signaling through the epidermal growth factor receptor (EGFR) leading ultimately to activation of an autocrine loop[15] (Figure 1).

*1.1 Mathematical Models of the Cell and Levels of Abstraction*

At the continuous level of abstraction, natural processes are described by detailed approaches drawn from the physical sciences involving continuous mathematics and analyzed using sophisticated numerical computation packages. However, while chemical or molecular events ultimately constitute biological processes, the complexity of these processes severely limits their accurate and effective description in terms of purely physical/chemical phenomena. This problem can be resolved at the discrete level of abstraction, where natural processes are described by purely symbolic expressions. Although this highly abstracted approach is an established means to analyze physical systems such as computer designs[16,17], it is also applicable to less predictable phenomena such as biological signaling processes. Indeed, biologists routinely reason about these processes at the discrete level, although this reasoning consists of informal notations and potentially ambiguous representations of important concepts like pathways, cycles, and feedback loops, with poor tool support. New rigorous but abstract models are needed for biology that: (i) accommodate conventional types of discrete reasoning based on experimentation, (ii) formally define a model and allowable reasoning steps, and (iii) provide predictive power for generating testable hypotheses.

Consider an analogy with algebraic analysis, such as the task of accurately and efficiently computing the polynomial $x^2$-$y^2$, given values for $x$ and $y$. One implementation of this task is based on the expression $(x+y)\times(x-y)$. Because this latter expression consists of two additions and only one multiplication it is inherently faster on many hardware platforms than the original expression, which requires one addition and two multiplications. To proceed with the implementation of this example, it would be necessary to decide whether the two expressions are equivalent. However, it is not apparent how many tests of equivalence would be sufficient to make this determination. A symbolic or formal methods approach to this task would be to assume a set of symbolic rewrite or inference rules and to reason algebraically. For example, starting from $(x+y)\times(x-y)$, it could be reasoned by the distributivity law that this expression is equal to $(x+y)\times x$ - $(x+y)\times y$. Again by distributivity it could be reasoned that the expression is equal to $x\times x + y\times x - x\times y - y\times y$. Using associativity, commutativity, and laws of subtraction, the $x\times y$ terms cancel, and it could be shown that the expression is equal to $x\times x$ - $y\times y$, and by the definition of exponentiation it could be shown that this is equal to $x^2$-$y^2$. Thus, subject to the validity of the axioms, it could be demonstrated that these polynomial expressions are symbolically equivalent **for all** $x$ and **for all** $y$. This type of reasoning is categorically different from numeric testing, but can be computationally challenging.

Major breakthroughs in efficient symbolic reasoning have occurred in the last decade of research in computer science. The framework of model checking[18,19], exponentially more efficient representations of Boolean and other functions[20], decision procedures, and efficient implementation of rewriting[21] represent quantum jumps in the ability to reason about symbolic systems, even when those systems may potentially have more states than there are atoms in the universe.

### *1.2 Pathway Logic*

Here we propose and describe *Pathway Logic*, an algebraic structure enabling the symbolic analysis of biological signaling pathways analogous to the standard definitions and laws for polynomials referred to above. We use the EGFR pathway as an example[15] in this discussion, but Pathway Logic could be applied to pathways regulating very diverse biological processes.

We use the Maude system to express the algebraic structure of Pathway Logic[13]. Maude is a high-performance reflective multiparadigm language and system which supports a wide range of applications. Maude implements rewriting logic, a logic of state and concurrent computation, and supports efficient logical reflection. This makes Maude remarkably extensible and powerful, and enables the creation of executable environments for different logics, languages, and models of computation, including abstract models of discrete biological computation.

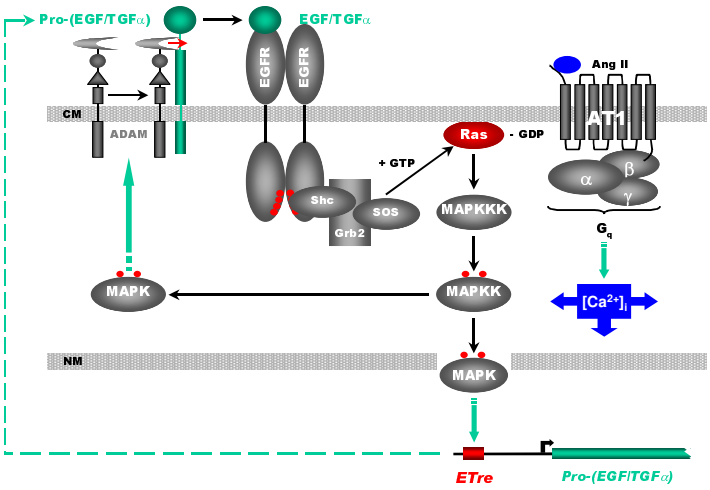


**Figure 1.** Fragment of the mammalian EGFR system illustrating activation of a downstream mitogenic signaling pathway involving the gene for the autocrine EGFR ligand TGFα. Also shown is a potential mechanism for cross-communication between the EGFR and a G protein-coupled receptor (AT1). Adapted from Gschwind et al.[15].

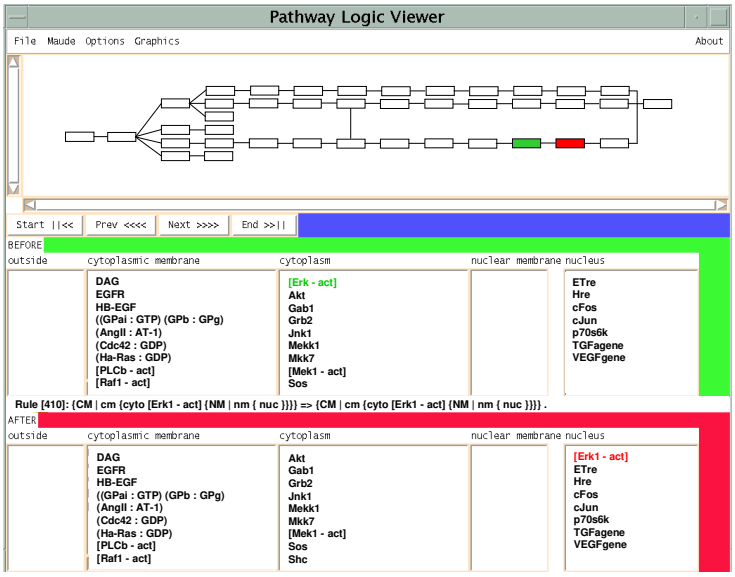


**Figure 2**. Screenshot of the current Pathway Logic viewer.

In what follows, numbered lines and bold font is used to highlight text written in the Maude executable specification language[13,14] to define the Pathway Logic algebra. The most basic sorts within the model presented here are termed **AminoAcid** and **Protein**. These sorts together with examples of their specific members, **S**, **T**, **Y** and **EGFR**, are declared by the following:

```
1  sorts AminoAcid Protein .
2  ops S T Y : -> AminoAcid .
3  op EGFR : -> Protein .
```

Statement 1simply declares the existence of the two sorts, **AminoAcid** and **Protein**. Statement 2 declares the amino acids serine **S**, threonine **T**, and tyrosine **Y** as constants of the sort **AminoAcid**, and statement 3 declares the **EGFR** as a constant of the sort **Protein**. The keywords **op** and **ops** are used to declare *operators* from a list of domain sorts into a range sort. In both of the examples above, the list of domain sorts is empty, indicating that the declared operators are *constants*.

Important ideas for the approach described here are explained using the **EGFR** pathway (Figure 1). These ideas are posttranslational protein modification, protein association, and cellular compartmentalization.

*Protein Modification.* We specify an algebra of protein modifications as follows:

```
4  sort Site Modification ModSet .
5  subsort Modification < ModSet .
6  op _ _ : AminoAcid MachineInt -> Site .
7  ops phos acetyl ubiq hydrox : Site -> Modification .
8  op none : -> ModSet .
9  op _ _ : ModSet ModSet -> ModSet [assoc comm id: none] .
10 op [_-_] : Protein ModSet -> Protein [right id: none] .
```

A site for modification on a protein is specified by a pair consisting of an amino acid and a machine integer joined by a binary juxtaposition operator (_ _) declared on line 6. On line 7 four operators are declared that represent the common protein modifications phosphorylation, acetylation, ubiquitinylation, and hydroxylation. Sets of modifications are formed by the subsort declaration (line 5) and an associative-commutative juxtaposition operator (line 9). Finally, sets of modifications are applied to proteins using the operator declared on line 10. Note that this operator has the empty set of modifications as its right identity. Thus, for any protein **P,** we have **[P - none] = P,** which means that the expression **[P - none]** is algebraically equivalent to **P**.

Like many signaling proteins, the EGFR is posttranslationally modified by phosphorylation[5]. Using this algebra, a common phosphorylation state of the EGFR can be modeled as follows:

11 **[EGFR - phos(Y 1092) phos(Y 1110) phos(Y 1172) phos(Y 1197)]**

This expression indicates that the modified EGFR is phosphorylated at tyrosines 1092, 1110, 1172, and 1197 (Y1092, Y1110, Y1172, and Y1197, respectively) after activation by EGF[22]. However, because there is only one known activation state of the EGFR, this expression can be simplified to:

12 **[FR – act]**

*Protein Association*. Signaling proteins commonly associate to form functional complexes[22]. This important phenomenon is algebraically represented by the following declarations:

13 **sort Complex .**
14 **subsort Protein < Complex .**
15 **op _:_ : Complex Complex -> Complex [comm] .**

That is, each protein is a singleton complex and two such complexes could be associated by the ":" operator on line 16 to obtain a multiprotein complex or module. Notice that this ":" operator has been declared commutative, but it is not assumed associative. Therefore, parentheses must be used to describe complexes formed from other complexes, such as the association shown on line 16 below between **EGF** and the **EGFR**.

16 **(EGF:([EGFR - act] : [EGFR - act]))**

*Protein Compartmentalization*. In the eukaryotic cell, proteins and other molecules exist in complex mixtures that are compartmentalized[2]. These compartmentalized mixtures (here termed "Soup" for convenience) are algebraically represented by the following declarations:

17 **sorts Soup Enclosure MemType.**
18 **subsort Complex < Soup .**
19 **op empty : Soup .**
20 **op _ _ : Soup Soup -> Soup [assoc comm id: empty] .**
21 **ops CM NM MM : -> MemType .**

22 **op {_|_{_}} : MemType Soup Soup -> Enclosure .**

An **Enclosure** is defined as a cellular membrane plus its **Soup**. A **MemType** denotes a specific membrane such as the cell membrane (CM) or the nuclear membrane (NM) (Figure 1). As with individual protein complexes (line 15), soups can also be combined as shown on line 20 by means of the binary soup union operator (with juxtaposition syntax). This union operator models the presumed fluid or dynamic nature of some subcellular compartments by specifying associative and commutative laws, so that no parentheses are needed and the order in which molecules exist in the soups does not matter.

*1.3 Rewriting Logic: Symbolic Modeling of Biochemical Reactions*

The algebraic structures or models described in Section 1.2 provide symbolic representations of protein modifications and eukaryotic cellular organization by means of an algebraic specification **S** involving sorts, subsorts, operators, and equational laws. To symbolize biochemical events such as signaling processes, we use theories in rewriting logic[23]. A rewrite theory is a pair **(S, R)** with **S** being an algebraic specification and **R** being a collection of rewrite rules. Each rewrite rule is of the form $\mathbf{l : t \Rightarrow t'}$, with **l** being a label, and **t** and **t'** being algebraic expressions in the algebra specified by **S**. Each rewrite rule specifies a local change or reaction that can occur in the system modeled by the theory **(S, R)**. These rewrite rules can precisely express biochemical processes or reactions involving single or multiple subcellular compartments. For example, the following text describes the first step in the activation of the EGFR signaling pathway, the binding of EGF to the EGFR[24] (Figure 1): **"Activated $MAPK^{erk1/2}$ is rapidly translocated to the nucleus where it is functionally sequestered and can regulate the activity of nuclear proteins including transcription factors."** In Maude syntax, this signaling process is described by the following rewrite rules:

23 **rl [410]: {CM | cm {cyto [Erk1 – act] {NM | nm {nuc}}}} =>**
**{CM | cm {cyto {NM | nm {nuc [Erk1 – act]}}}} .**
24 **rl [436B]: [Erk1 – act] ETre => [Erk1 – act] [ETre – act] .**

In Figure 1, MAPK represents ERK1 and ETre is a consensus DNA binding site for transcription factors that activate expression of the *TGFα* gene. Notice that in the new state of the system represented by the right hand side of rule 410, activated ERK1 is present in the nucleus following translocation. The right hand side of rule 436B indicates that activated ERK1 has induced transcription of the *TGFα* gene through the ETre element in its 5′-regulatory region. These rewrite rules describe a local change that could occur when an instance of the left-hand side of each rule

exists in a cell. Mathematically, the rules in **R** are applied modulo the equivalence between expressions defined by algebraic laws in **S**.

In a system specified by a rewrite theory **(S, R)**, rewriting logic allows reasoning about *possible* complex changes given the basic changes specified in **R**, such that any such change is possible if and only if it can be proved to be derived using the rules in $\mathbf{R}^{14}$. These complex changes can be concurrent; that is, different parts of a compartment can change simultaneously and independently. Furthermore, the changes can be multi-step, allowing reasoning about *possible future* states of the system. Thus, under very reasonable assumptions rewrite theories can be executed in Maude to describe a biological signaling process over time according to a symbolic model, and can be formally analyzed to reason about properties of the states reachable from an initial state.

## 2 Analysis Techniques

Given a formal symbolic model of some part of the signaling pathways in a cell as a rewrite theory, several kinds of automatic analysis can be performed. We note that each biochemical reaction that is represented as a rewrite rule conserves proteins. This conclusion has two important consequences for the term rewriting system that simplify its analysis. First, from a given initial state the set of reachable states is finite. Second, each Soup variable occurring in the left-hand side must also occur in the right-hand side, as otherwise there would be destruction of arbitrary numbers of molecules bound to the variable.

### *2.1 Static Analysis*

As a prelude to analyzing the dynamic behavior of a model, we can first perform static analysis. As one example, just considering the simplest algebraic part **S** of the model (without the rewrite rules **R**), we can identify sorts that are inhabited only by declared constants. Typically, these sorts will capture the notion of a family of related proteins (the constants). A rule containing variables ranging over such sorts is (for the purposes of ground rewriting) equivalent to a set of rules obtained by instantiating such variables with the constants inhabiting their sorts in all possible ways.

### *2.2 Forward and Backward Search*

The simplest form of analysis of the dynamic behavior of a model is to *run* the model from a given initial state by applying the rules in an arbitrary order for some

fixed number $n$ of rewrites or until no more rules are applicable. In Maude, this process is done with the *rewrite* command. Since most models will be nondeterministic and the future states reachable from a given initial state form a graph where paths diverge, converge, or cycle back on themselves, the *search* command in Maude supports a breadth-first search through this transition graph looking for states that match some pattern, possibly with a side condition. Using search, all possible outcomes can be identified from a given initial state. When the rules are unconditional and each variable that occurs in the left-hand side occurs in the right-hand side it is possible to flip the rules over and run the model backwards – either as a simulation or as a search. Using such reversed rules, we can ask questions of the form "from what initial state(s) can we get to some desired (or known) final state?"

*2.3 Explicit state Model Checking*

The search command allows us to examine the transition graph produced from a given initial state for states satisfying some static property, such as the existence of a protein in some particular phosphorylation state. We may want to ask more complex queries about the paths in the transition graph, such as "if we reach a state that satisfies property **P**, is it true that we must eventually reach a state that satisfies property **Q**?" We may also want to restrict our attention to the subset of paths in the transition graph that satisfy some *fairness* criterion such as "if reaction **R** is always possible, then eventually reaction **R** happens".

A language suitable for framing such queries is propositional linear temporal logic (LTL). Here the propositions correspond to properties that can be statically checked for each state. The familiar propositional calculus with its operators such as $\wedge$ ("and"), $\rightarrow$ ("implies"), and $\neg$ ("not"") is extended with temporal operators such as $\Box$ ("always") and $\Diamond$ ("eventually"). Standard techniques based on Büchi automata can be used to check if the transition graph produced from a given initial state satisfies an LTL formula[17].

*2.4 Meta-analysis*

In practice there may be uncertainty regarding the experimental evidence for certain reactions. Thus, we consider a parameterized specification that describes a finite family of models. Rather than a fixed set **R** of reactions, we have some base set $\mathbf{R}_{\text{base}}$ of reactions about which we are confident, together with a set $\mathbf{P} = \{P_1,\ldots, P_n\}$ of parameters (often $n$ will be 1) and for each parameter $P_i$, a set $\{A_{i,1},\ldots, A_{i,m_i}\}$ of alternative instantiations. Here each $A_{i,j}$ is a set of reactions. Such a specification describes a family of $\prod_{i=1}^{n} m_i$ distinct models, which are obtained by choosing

different combinations of instantiations for $P_1,\ldots, P_n$ and adjoining them to $\mathbf{R}_{base}$. Given a LTL formula and an initial state we can now search this family of models to find those models for which an LTL formula is true in the initial state.

*2.5 Key Benefit: Expressive Questions*

The application of model checking and other analysis techniques is important for the expressiveness of the questions for which answers can be effectively computed. In particular, providing a complete search of the space of all possible executions of an abstraction of system has been found in other domains to be more useful than the forward simulation (testing) of just some possibilities for that system. The complete symbolic exploration of all reaction interactions can provide useful insights, and can directly enable a biologist to ask questions such as "If EGF is not present to stimulate it's pathway, but angiotensin II is, is the ERK signal silent?" (Figure 1). This kind of expressive question can be directly encoded in temporal logic as follows: $\Box(\Box(\text{AngII} \wedge \neg\text{EGF}) \rightarrow \neg\Diamond\text{ERK1})$. The answer to such queries (including traces demonstrating counterexamples) can be effectively computed using the techniques described above. Thus far, we have encoded hundreds of reactions relating to signaling pathways in mammalian cell cycle control, and we have computed all possible outcomes from certain interesting states using the Maude search command.

**3. Graphical Representation for Pathway Logic**

To make Maude-generated models easily accessible to a user we have developed a graphical viewer tool. The representation is a directed transition graph, with nodes representing the states of proteins within a model and transitions representing rewriting with respect to applicable rules. The state of the system is specified by the contents of each of its compartments. One possible realization of the viewer is shown in Figure 2. In the top part of the graphical user interface is a canvas displaying the directed transition graph constructed from Maude generated paths or traces. Circles depict the states, and arrows indicate the state transitions. Every transition is associated with the rewriting operation with respect to a certain rule, and the rules are shown below the arrows. Below the canvas is a set of boxes that specify the Soups before and after a rewriting transition. In this example the Maude output from a simple search operation shows potential crosstalk between known pathways. The user has selected a step in the pathway to examine in detail, with the green before state and red after state displayed in detail, and highlighted in the pathway overview.

## 4. Summary

Our major hypothesis is that useful computational analysis can be performed on biological signaling networks at a very high level of abstraction.

We have presented an approach of applying formal methods to the analysis of biological pathways. Most previous related work has focused on continuous models of such systems, an approach restricted by lack of detailed *in vivo* rate and concentration data and the computational complexity of simulations at that level of abstraction. Some previous work such as that on EcoCyc and PiFPC has begun to show the benefits of higher level abstractions[26,12]. The PiFPC work is progressing toward stochastic and rate-based models within a formal framework. Here we propose a formal framework and the application of modern model checking and other symbolic techniques to signaling networks at the higher level of abstraction, but enabling the answers to queries of a different nature than simple forward simulation.

*Future Work.* In the near future, we will explore the automated connection between different levels of abstraction of biological modeling. Using automated symbolic abstraction methods we will be able to simultaneously represent low-level molecular details, and higher-level protein module-at-a-time or compartment-at-a-time structure, thus enabling scalable computational analysis of extremely large systems. We will also begin to symbolically represent delays, which will begin to allow reasoning about circadian and other cellular rhythms, without reliance on detailed *in vivo* rate data. We will also experiment with perturbations of the pathways, computing possible outcomes of induced signal or network changes. Finally, we will use temporal logic specifications to study the property-sensitive differences between related pathways.

## Acknowledgements

We thank Tom Garvey, Peter Karp, Pedro Romero, Raymonde Guindon, Andrea Lincoln, Marianna Yanovsky, Natarajan Shankar for helpful discussions, and the anonymous reviewers for their comments.

## References

1. O.G. Vukmirovic and S.M. Tilghman. Exploring genome space. *Nature*, 405:820-822, 2000.

2. J.D. Jordan, E.Landau, and R. Iyengar. Signaling networks: The origins of cellular multitasking. *Cell*, 103:193-200, 2000.
3. K.W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol Biol Cell*, 10:2703-2734, 1999.
4. D. Fambrough, K. McClure, A. Kazlauskas, and E.S. Lander Diverse signaling pathways activated by growth factor receptors induce broadly overlapping, rather than independent, sets of genes. *Cell*, 97:727-741, 1999.
5. T. Pawson and T.M. Saxton. Signaling networks--do all roads lead to the same genes? *Cell*, 97:675-678, 1999.
6. K.W. Kohn. Functional capabilities of molecular network components controlling the mammalian G1/S cell cycle phase transition. *Oncogene*. 16:1065-75, 1998.
7. G. Weng, U.S. Bhalla, and R. Iyengar. Complexity in biological signaling systems. *Science*, 284:92-96, 1999.
8. H.H. McAdams and A. Arkin. It's a noisy business! Genetic regulation at the nanomolar scale. Trends Genet, 15:65-69, 1999.
9. P.D. Karp, M. Riley, S.M. Paley, A. Pellegrii-Toole, and M. Krummenacker. Ecocyc: Encyclopedia of Escherichia coli genes and metabolism. *Nucleic Acids Res*, 26(1):50-53, 1998.
10. C.A. Ouzounis and P.D. Karp. Global properties of the metabolic map of Escherichia coli. *Genome Res*, 10(4):568-576, 2000.
11. P.R. Romero and P. Karp. Nutrient-related analysis of pathway/genome databases. In R.B. Altman et al., editor, Pacific Symposium on Biocomputing 2001, pages 471-482. World Scientific, 2001.
12. Aviv Regev, William Silverman, and Ehud Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In R.B. Altman et al., editor, Pacific Symposium on Biocomputing 2001, pages 459-470. World Scientific, 2001.
13. Manuel Clavel, Francisco Duran, Steven Eker, Patrick Lincoln, Narciso Marti-Oliet, Jose Meseguer, and Jose Quesada. A tutorial on Maude. SRI International, March 2000, http://maude.csl.sri.com.
14. Manuel Clavel, Francisco Duran, Steven Eker, Patrick Lincoln, Narciso Marti-Oliet, Jose Meseguer, and Jose Quesada. Towards Maude 2.0. In K.Futatsugi, editor, Proc. 3rd. Intl. Workshop on Rewriting Logic and its Applications, volume 36 of ENTCS Elsevier, 2000.
15. A. Gschwind, E. Zwick, N. Prenzel, M. Leserer, and A. Ullrich. Cell communication networks: epidermal growth factor receptor transactivation as the paradigm for interreceptor signal transmission. *Oncogene,* 20:1594-1600, 2001.
16. Steven P. Miller and Mandayam Srivas. Formal verification of the AAMP5 microprocessor: A case study in the industrial use of formal methods. In WIFT

'95: Workshop on Industrial-Strength Formal Specification Techniques, pages 2-16, Boca Raton, FL, 1995. IEEE Computer Society.

17. Ed Clarke, Orna Grumberg, and Doron Peled. Model Checking. MIT Press, 1999.
18. J.P. Queille and J. Sifakis. Specification and verification of concurrent systems in Cesar. In Proceedings of the 5th International Symposium on Programming, volume 137 of Lecture Notes in Computer Science, pages 337-351, Turin, Italy, April 1982. Springer-Verlag.
19. 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, April 1986.
20. R.E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677-691, August 1986.
21. Manuel Clavel, Francisco Duran, Steven Eker, Patrick Lincoln, Narciso Marti-Oliet, Jose Meseguer, and Jose Quesada. The Maude System. In P. Narendran and M.Rusinowitch, editors, Procs. 10th Intl. Conference on Rewriting Techniques and Applications (RTA-99), volume 1632 of Lecture Notes in Computer Science, pages 240-243. Springer-Verlag,1999.
22. W. Kolch. Meaningful relationships: The regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *Biochem J*., 351:289-305, 2000.
23. Jose Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73-155, 1992.
24. J. Schlessinger. Cell signaling by receptor tyrosine kinases. *Cell*, 103:211-225, 2000.
25. Peter D. Karp. Pathway Databases: A Case Study in Computational Symbolic Theories. *Science* 293(5537): 2040-2044, Sept 2001.