Pathway Logic Modeling of Protein Functional Domains in Signal Transduction

Carolyn Talcott, Steven Eker, Merrill Knapp, Patrick Lincoln, Keith Laderoute
SRI International, 333 Ravenswood Avenue, Menlo Park CA 94025
{*firstname.lastname*}*@sri.com*

Abstract

Protein functional domains (PFDs) are consensus sequences within signaling molecules that recognize and assemble other signaling components into complexes. Here we describe the application of an approach called Pathway Logic to the symbolic modeling signal transduction networks at the level of PFDs. These models are developed using Maude, a symbolic language founded on rewriting logic. Models can be queried (analyzed) using the execution, search and model-checking tools of Maude. We show how signal transduction processes can be modeled using Maude at very different levels of abstraction involving either an overall state of a protein or its PFDs and their interactions. The key insight for the latter is our algebraic representation of binding interactions as a graph.

1 Introduction

There is a practical need to represent very large biological networks of all kinds as models at different levels of abstraction. For example, consider the following:

- The proteome of eukaryotic cells is at least an order of magnitude larger than the genome (very large and diverse protein networks)
- A large fraction of the genome of mammalian cells ($\approx 10\%$ of the human genome) encodes genomic regulators producing very large regulatory networks of the genome itself
- Biological networks interact as modules/subnetworks to produce high levels of physiological organization (e.g., circadian clock subnetworks are integrated with metabolic, survival, and growth subnetworks)

In silico models of such networks would be valuable but must have certain features. In particular, they must be easily modified–extended or updated–and useable by bench researchers for formulating and testing hypotheses about how signals and other changes are propagated.

Pathway Logic [1,2] is an application of techniques from formal methods and rewriting logic [3] to develop models of biological processes. The goals of the Pathway Logic work include: building network models that working biologists and biomedical researchers can interact with and modify; making formal methods tools accessible to the general biological and biomedical research community; and enabling wet-lab researchers to generate informed hypotheses about complex biological networks.

The Pathway Logic work has initially focused on curation of models of signal transduction networks, including the Epidermal Growth Factor Receptor (EGFR) network and closely related networks [4,5,6]. Signal transduction processes are modeled at different levels of abstraction involving: (I) the overall state of proteins, or (II) protein functional domains (PFDs) and their interactions. These signaling networks can be queried using formal methods tools, for example, by choosing an initial condition and trying the following: (i) execution—show me some signaling pathway; (ii) search—show me all pathways leading to a specified final condition; or (iii) model-checking—is there a pathway with certain given properties?

In this paper we use the recruitment and activation of the ubiquitous Raf1 serine-threonine protein kinase to illustrate the two levels of representation and in particular to show how PFDs are modeled and how the resulting model can be used. This more detailed representation of signaling proteins in which PFDs are explicit can be used to model domain specific interactions in signaling networks, an important area of modern signal transduction research. Future work includes expanding the collection of proteins modeled at the level of PFD interactions as data becomes available, modeling additional signal transduction networks and modeling metabolic pathways and their interactions with signal transduction pathways.

1.1 Formal Methods in Biology

Formal methods techniques have been used by various groups to develop executable models of biological systems at high levels of abstraction. Typically the techniques are based on a model of concurrent computation with associated formal languages for describing system behavior and tools for simulation and analysis.

Petri nets were developed to specify and analyze concurrent systems. There are many variants of the Petri net formalism and a variety of languages and tools for specification and analysis of systems using the Petri net model [7]. Petri nets have a graphical representation that corresponds naturally to conventional representations of biochemical networks. They have been used to model metabolic pathways and simple genetic networks (examples include [8,9,10,11]). However, these efforts have largely been concerned with kinetic or stochastic models of biochemistry. In [12] a more abstract and qualitative view was taken, mapping biochemical concepts such as stoichiometry, flux modes, and conservation relations to well-known Petri net theory concepts.

The pi-calculus [13] is a process algebra originally developed for describing concurrent computer processes. There are a number of specification languages and tools based on the pi-calculus. A pi-calculus model for the receptor tyrosine kinase/mitogen-activated protein kinase (RTK/-MAPK) signal transduction pathway is presented in [14]. Signaling proteins are represented as processes and interactions as synchronous communications between processes (handshakes).

A stochastic variant of the pi-calculus is used in [15] to model both the time and probability of biochemical reactions.

Statecharts are a visual notation for specifying reactive concurrent systems[16] used in object-oriented software design methodologies. Statecharts naturally express compartmentalization and hierarchical processes as well as flow of control amongst subprocesses. The resulting models can be used for simulation and visualization of biochemical processes. Statecharts have been used to model biological processes such as T-cell activation[17,18].

Live Sequence Charts [19] are an extension of the Message Sequence Charts modeling notation for system design. Using the associated PlayIn/PlayOut approach, models can be built and tested by acting out reaction scenarios. Models of subsystems can be combined and charts can be annotated with assertions that allow invariants and prohibited conditions to be expressed and checked. This approach has been used to model the process of cell fate acquisition during C.elegans vulval development[20].

1.2 Pathway Logic

Pathway Logic is an approach to modeling biological entities and processes based on formal methods and rewriting logic [3]. Pathway Logic models are developed using the Maude (`http://maude.csl.sri.com`) system, a formal language and tool set based on rewriting logic.

Like the approaches to modeling biological processes mentioned above, Pathway Logic models are executable—hence they can be used for simulation. In addition, the Maude system provides search and model-checking capabilities. Using the search capability all possible future states of a system can be computed to show its evolution from a given initial state (specified by the states of individual components) in response to a stimulus or perturbation. Using model-checking a system in a given initial state can be shown to never exhibit pathways with certain properties, or the model-checker can be used to produce a pathway with a given property (by trying to show that no such pathway exists). Using the reflective capability of Maude, models can be mapped to other formalisms and exported in formats suitable for input to other tools for additional analysis capabilities and visualization.

Rewriting Logic[3], is a logical formalism based on two simple ideas: states of a system are represented as elements of an algebraic data type; and the behavior of a system is given by local transitions between states described by abstractions called rewrite rules. In Pathway logic, algebraic data types are used to represent concepts from cell biology needed to model signaling processes, including intracellular proteins, biochemicals such as second messengers, extracellular stimuli, biochemical modification of proteins, protein association, and cellular compartmentalization of proteins. Rewrite rules are used to model local processes within a cell or transmission of a signal across a cell membrane. A signaling network is represented as a collection of rewrite rules together with the algebraic decla-

rations. Rewriting logic then allows reasoning about possible complex changes given the basic changes (rules) specified by the model. In particular, pathways in the network satisfying different properties can be generated automatically using tools based on logical inference for execution (deduction), search, and model-checking.

2 Activation of Raf1 modeled at two levels

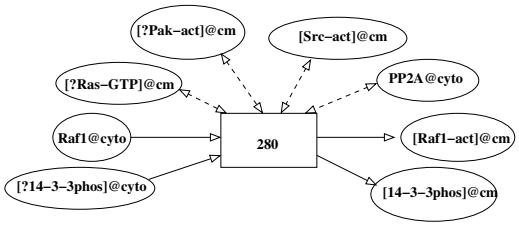
A Pathway Logic model of the Epidermal Growth Factor Receptor (EGFR) network (reviewed in [4,5,6]) is being developed by curating rewrite rules for relevant biochemical processes from the scientific literature. Depending on what data is available, processes are modelled at different levels of abstraction. *Level I* rules model processes in terms of overall protein states. Protein functional domains (PFDs) are consensus sequences within signaling molecules that recognize and bind other signaling components to make complexes. When there is enough information about a protein and the domains it contains to hypothesize the details of activation and translocation *Level II* rules are developed. These rules model processes in terms of protein functional domains and explicit posttranslational modifications of individual signaling molecules are included in the model. A key idea for the Level II rules is the representation of PFDs and their interactions algebraically as a graph.

Here we use the recruitment and activation of the ubiquitous Raf1 serine-threonine protein kinase to illustrate the two levels of representation. The Raf1 system is a reasonably well-established and detailed example of a signal integrator in the EGFR network [21,22]. The Raf1 kinase is an effector of EGFR and other RTK signaling through the ERK1/2 MAPK pathway, which is organized in a module that can be represented by the kinase cascade MAPKKK $\Rightarrow$ MAPKK $\Rightarrow$ MAPK (reviewed in [5]). In this module, Raf1 is a MAPKKK.

2.1 Activation of Raf1 at Level I

An early step in the activation of Raf1 is recruitment of cytoplasmic Raf1 to the inner side of the cell membrane by Ras, following stimulation of the EGFR. Figure 1 shows both a graphical representation and the Maude representation (from which the picture is generated) of the Level I rule 280 modeling the activation of Raf1 and its recruitment to the cell membrane. This rule says that if the cell contains a Ras type protein with a GTP modification, activated Pak, and Src protein kinases on the interior side of the cell membrane, and Raf1, phosphorylated 14-3-3 scaffold/adaptor proteins, and the phosphatase PP2A in the cytoplasm, then Raf1 can be activated and recruited to the membrane along with 14-3-3, leaving PP2A in the cytoplasm.

In Maude a cell is represented by a term of the form `{CM | ... { ... }}` where the first ellipsis stands for biochemicals in or attached to the interior of the




```
crl[280.?Ras.?Pak.Src.PP2A.?14-3-3.->.Raf1]:
   {CM | cm [?Ras - GTP] [?Pak - act] [Src - act]
        {cyto Raf1 [?14-3-3 - phos] PP2A }}
   =>
   {CM | cm [?Ras - GTP] [?Pak - act] [Src - act]
         [Raf1 - act] [?14-3-3 - phos] {cyto PP2A}}
 if ?Ras S:Soup := N-Ras K-Ras H-Ras
 [metadata "21192014(R)"]  .
```

Figure 1: Raf1 activation rule (Level I)

cell membrane, and the second ellipsis stands for the biochemicals and compartments in the cytoplasm. A particular cell state is represented by replacing the ellipses by terms representing specific biochemicals and compartments. In a Maude rule the ellipses are replaced by patterns—terms with variables ranging over some set of biochemicals, represented as *sorts* in Maude. One of the sorts is `Ras` representing the Ras type proteins. We use the convention that the name of a class of proteins prfixed by a `?` is a variable ranging over the corresponding sort. Thus `?Ras` can be instantiated to any of the proteins in the model declared to be of sort `Ras`. At Level I, posttranscriptional modification is represented abstractly by a modification operator `[_-_]` applied to a protein and a set of abstract modifications. In the left-hand side of rule 280 the term `[?Ras - GTP]` represents a Ras type protein with a GTP modification, while the term `[Src - act]` represents activated `Src` protein kinase on the interior side of the cell membrane. The occurrence of `Raf1`, `PP2A`, and `[?14-3-3 - phos]` represent Raf1, PP2A and phosphorylated 14-3-3 in the cytoplasm. The variables `cm` and `cyto` serve a place holders for any remaining unspecified biochemicals in (or on the interior side of) the cell membrane, and the cytoplasm respectively.

In order to apply a set of rules to a particular cell, the components of that cell are formally represented as a multiset of ground terms (constants and other terms containing no variables) declared to be the initial cell state. A rule such as 280 is then applied to the cell by finding a substitution of components for the variables appearing in the left-hand side that make it equal to the cell in question (matching), and replacing the cell by the result of applying the matching substitution

to right-hand side of the rule. Representing cell contents using multisets means that the order that individual components are listed in does not matter, and the matching process takes this into consisderation. With the above in mind we can see that application of rule 280 to the initial cell state:

```
eq cell = PD({CM | [N-Ras - GTP] [Pak1 - act] [Src - act]
           {Raf1 [14-3-3t - phos] PP2A }}) .
```

does indeed move Raf1 and 14-3-3 from the cytoplasm to the membrane, activating Raf1 and leaving the phosphorylation state of the 14-3-3 protein unchanged.

The condition following the `if` in rule 280 constrains the matching protein found for the variable `Ras` to be one of those listed. The term

```
[metadata "21192014"]
```

represents information that is not used in execution of the model but provides evidence and other useful information that can used in other operations on the model. This particular metadata is the medline citation for a paper used in curation of the rule.

Level I rules have an alternative representation in terms of *occurrence*s and *transition*s (corresponding to a special kind of Petri net), An occurrence is a biochemical paired with its location in the cell. For example, the occurrence of Raf1 on the left hand side of the rule is represented by the pair `< Raf1, cyto >` and the pair `< [Raf1 - act], cm >` represents the occurrence on the right-hand side. A rule is then represented by a triple consisting of the multiset of left-hand side occurrences, the rule identifier, and the multiset of right-hand side occurrences. (Generic variables such as `cm` and `cyto` are ignored.) In the picture the occurrences are represented by ovals labelled by a printed form and the transistion by a rectangle labeled with the rule identifier. Occurrences that appear only on the left-hand side are indicated by arrows from the oval to the rectangle, those that appear only on the right-hand side by arrows from the rectangle to the oval, and those that appear on both sides (enzymes, coenzymes) by dashed bidirectional arrows.

2.2 Activation of Raf1 at Level II

The difference between a Level I rule and a Level II rule is that a Level I rule deals with interactions between whole proteins whereas a Level II rule deals with interactions between protein domains. In Level I, Raf1 is considered to be inactive by (1) not having the modification ”act” and (2) being located in the cytoplasm. In Level II the phosphorylation states of relevant amino acids, the domains and sites which are bound intra- or inter-molecularly are made explicit.

Based on work by Dhillon and Kolch[22] (augmented with details from a number of other publications) we drew, by hand, a stylized diagram of a possible Raf1 activation process (Figure 2). The diagram is focused on the Raf1 protein. Raf1 is represented as a list of domains (blue bars) and potential phosphorylation sites

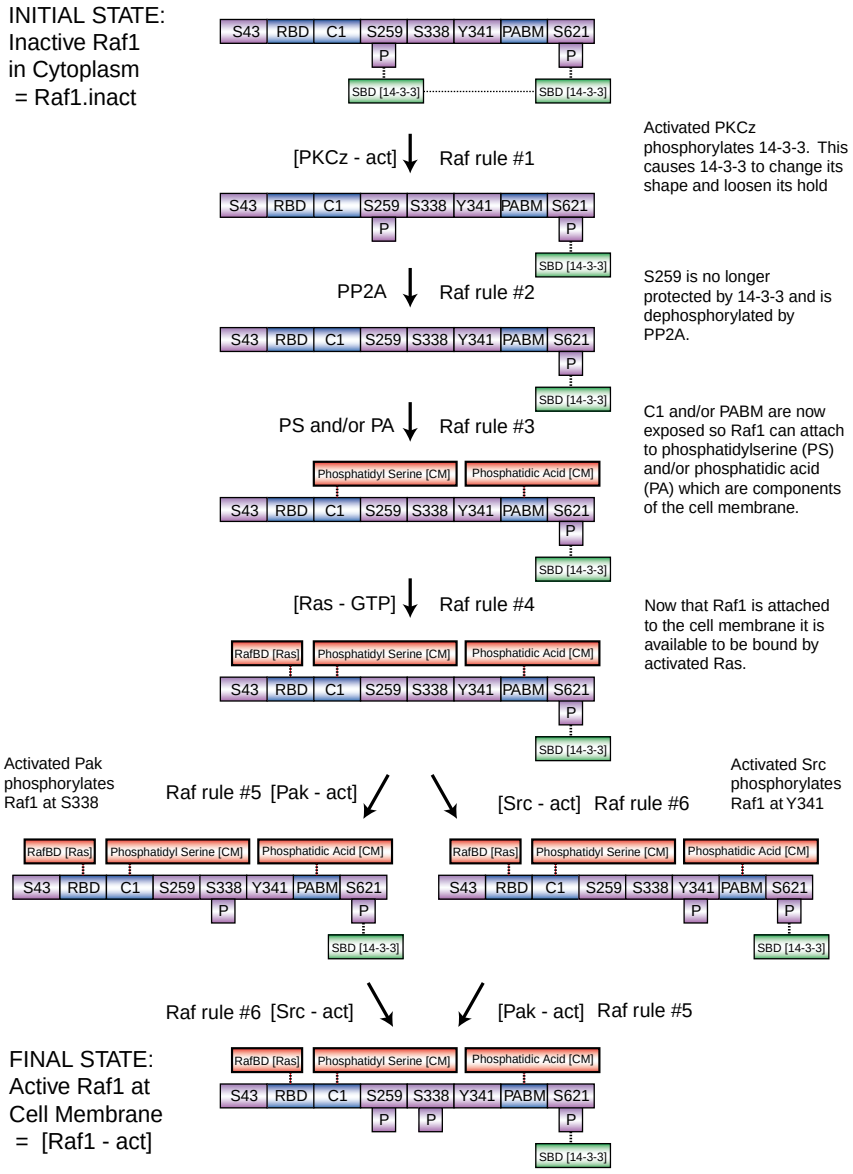


Figure 2: Raf activation (Level II)

(lavender bars) relevant to the interaction being studied. Phosphorylation is indicated by a button labeled P hanging below the site bar. Other proteins binding to Raf1 are represented by a bar labeled by the bound domain and the protein name. Those above the Raf1 list (red) are in or attached to the cell membrane (also indicated by [CM]), and those below (green) are in the cytoplasm. The first row of the diagram represents inactive Raf1. It is associated with a dimer of 14-3-3 scaffold/adaptor proteins through binding of phosphorylated serines 259 and 621 in Raf1 to serine binding domains (`SBD`) in the 14-3-3 dimer. In the diagram the 14-3-3 dimer is represented by the two 14-3-3 binding domains (green bars) and the line connecting these domains to each other.

The arrows in the diagram indicate the progression of the activation process and the arrow labels give a description of the rule governing the interaction and indicate the key triggering biochemistry. For example, the trigger for Raf rule `#1` is activated PKCz (`[PKCz - act]`).

Based on this diagram, rules were written to model the steps of Raf1 activation. To represent the functional domains of a signaling protein explicitly, we annotate proteins using the notation `[ p:Protein | atts:Atts ]`. Here `atts:Atts` is a set of attributes representing one or more PFDs or amino acid residues (sites). Each attribute may have associated modifications such as phosphorylation (`phos`) or an indication that the domain/site is participating in a binding (`bound`). Thus, a protein at Level II can be thought of as an encapsulated collection of functional domains and sites. The association or binding of signaling proteins through their functional domains is explicitly represented by edges in a graph whose nodes are protein-attribute pairs. For example the inactivated form of Raf1 shown in the first row of Figure 2 is represented by right-hand side of the following Maude equation.

```
eq Raf1.inact =
   [Raf1 | (S 43), RBD, C1, (S 259 - phos - bound), (S 338),
           (Y 341), PABM, (S 621 - phos - bound)]
   [14-3-3a | (SBD - bound), (DMD - bound)]
   [14-3-3b | (SBD - bound), (DMD - bound), (T 141)]
   e((Raf1,(S 621)), (14-3-3a,SBD))
   e((Raf1,(S 259)), (14-3-3b,SBD))
   e((14-3-3a,DMD),  (14-3-3b,DMD)) .
```

The attributes

```
  (S 43), RBD, C1, (S 259 - phos - bound), (S 338),
  (Y 341), PABM, (S 621 - phos - bound)
```

correspond to the bars in Figure 2. The attribute `(S 621 - phos - bound)` denotes the site `(S 621)` with two modifications `phos` and `bound`. The modifications `-phos` on the sites `S 259` and `S 621` correspond to the buttons labeled P and the modification, `-bound` is used to indicate locally that the attribute has a binding. In the Maude term the 14-3-3 dimer is represented by the two 14-3-3 protein terms, and the edge `e((14-3-3a,DMD),(14-3-3b,DMD))`

The two vertical lines connecting the phosphorylated sites on Raf1 to the 14-3-3 dimer are represented in the Maude term by the edges

```
e((Raf1,(S 621)), (14-3-3a,SBD))
e((Raf1,(S 621)), (14-3-3a,SBD)) .
```

In the Level II representation the activation of Raf1, represented at Level I by the single rule 280, requires several rules in which structural features of some of the proteins, including Raf1, are annotated with information about relevant PFDs and binding sites, and the binding between proteins is made explicit. As an example, we show the Maude representation of the rule numbered 6 in the diagram, in which activated Src phosphorylates partially activated Raf1 at Tyrosine 341.

```
rl[Raf1#6.Y341phos]:
 {CM | cm PS PA [?Slk - act]
  [?Ras | GTPbound, (RafBD - bound)]
  [Raf1 | (S 43), (S 259), (Y 341), (C1 - bound),
          (S 621 - phos - bound), (PABM - bound),
          (RBD - bound), raf1:Atts]
  [14-3-3a | (SBD - bound),(DMD - bound),1a:Atts]
  [14-3-3b |  SBD, (DMD - bound), (T 141 - phos)]
  e((14-3-3a, DMD), (14-3-3b,DMD))
  e((Raf1, (S 621)), (14-3-3a,SBD))
  e((Raf1, C1), b(PS)) e((Raf1, PABM), b(PA))
  e((Raf1, RBD), (?Ras, RafBD)) {cyto}}
=>
 {CM | cm PS PA [?Slk - act]
  [?Ras | GTPbound, (RafBD - bound)]
  [Raf1 | (S 43), (S 259), (Y 341 - phos),
          (S 621 - phos - bound), (PABM - bound),
          (C1 - bound), (RBD - bound), raf1:Atts]
  [14-3-3a | (SBD - bound),(DMD - bound),1a:Atts]
  [14-3-3b | SBD,(DMD - bound), (T 141 - phos)]
  e((14-3-3a, DMD), (14-3-3b,DMD))
  e((Raf1, (S 621)), (14-3-3a,SBD))
  e((Raf1, C1), b(PS))  e((Raf1, PABM), b(PA))
  e((Raf1, RBD), (?Ras, RafBD)) {cyto}} .
```

The left-hand side of rule matches a situation in which Raf1 is associated with a dimer of 14-3-3 proteins through binding of phosphorylated serine 621 (represented by `(S 621 -phos - bound)`) to the serine-binding domain (`(SBD - bound)`) in the 14-3-3 dimer, represented by the edge

`e((Raf1,(S 621)), (14-3-3a,SBD))`.

The additional requirements that Raf1 must be bound to Ras, phosphotidylserine (PS), and phosphatidic acid (PA) are represented by the edges

```
e((Raf1, C1), b(PS)) e((Raf1, PABM), b(PA))
e((Raf1, RBD), (?Ras, RafBD))
```

where the terms `b(PS)` and `b(PA)` represent unspecified binding domains or sites on PS and PA respectively. Notice that the representation of overall cell structure is the same and that Level I and Level II notation for proteins can be mixed, only using Level II detail where relevant. For example, Src is used as a Level I protein (as a variable `?Slk`) of sort `Slk` (Src like kinase).

In order for Raf1 to be fully activated it must be phosphorylated on both Y341 (by a Src-like-kinase) and on S338 (by a member of the Pak family). It is unclear whether Y341 or S338 is phosphorylated first. This is represented in Figure 2 by the branch in the sequence of rules. In the Maude representation, rule 6 deals with this ambiguity by using the variable `raf1:Atts` instead of requiring a particular phosphorylation state for S338. Rule 5 (not shown) similarly uses an attribute variable instead of requiring a particular phosphorylation state for Y341.

The application of Level II rules follows the same procedure as for Level I. Although domains and sites have a fixed order within a protein sequence, in the Maude model we treat them as a set because the ordering information plays no role in the processes represented. (Some ordering information is implicit in the site numbers and could easily be added if required for other purposes.)

Level II rules for Raf1 are connected to Level I by the equational rule shown above that converts the Level I representation `Raf1.inact` of inactivated Raf1 to its Level II representation, and a dual rule that converts the Level II complex representing activated Raf1 to its Level I representation (rule 7 in the pathway shown below).

3 Using the Pathway Logic Model

We now illustrate some of the ways in which the tools supplied by Maude can be used to query and analyze a Pathway Logic model. To set a context for using the rules for Raf1 activation at the PFD level (Level II) we define an initial cell state (`qraf` containing inactive Raf1 and postulated necessary conditions to activate it.

```
  eq qraf =  PD({CM | PS PA [Pak1 - act] [PKCz - act]
                      [Src - act] [H-Ras - GTP]
                      {Raf1.inact PP2A}} ) .
```

The form `PD( ... )` represents a cell in a Petri dish, possibly with some external signaling compounds. As a first example of using the model, the question "can Raf1 in a cell described by `qraf` be activated?" is answered by defining a proposition `praf0` that expresses the query and then using the `findPath` query.

```
eq PD( out {CM | cm [Raf1 - act] {cyto}} )
   |= praf0 = true .
```

The above equation says that the proposition `praf0` is true for a cell if the dish containing it matches the pattern on the left .

The query `findPath(qraf,praf0)` uses the Maude model checker to find a counter example to the assertion that no state satisfying `praf0` can be reached from the initial state `qraf` by applying the rules of the model (in this case the equation for `Raf1.inact` and Raf rules 1-7). If a counter example is found, the query function extracts a path giving the labels of rules applied and the state reached that satisfies the property `praf0`. The Maude command `red findPath(qraf,praf0)` executes this query, returning the following.

```
result SimplePath:
spath('Raf1#1.PKCz 'Raf1#2.PP2A 'Raf1#3.PS.PA 'Raf1#4.Ras
     'Raf1#5.S338phos 'Raf1#6.Y341phos 'Raf1#7.Raf1.is.act,
      PD({CM | PA PS [Pak1 - act] [PKCz - act] [Raf1 - act]
      [H-Ras - GTP] [Src - act] {14-3-3b PP2A 14-3-3a}}))
```

The label `Raf1#7.Raf1.is.act` refers to a rule that converts the Raf1 complex from Level II to Level I to connect with downstream Level I rules.

To determine if other pathways are possible, we use the search command

```
search qraf =>! d:Dish .
```

to ask for all paths leading to a final state (a state to which no more rewrite rules apply). The answer here is that there is one final state, the one found by the above query, and two paths. The second path differs from the first only in the order in which rules 5 and 6 are applied. In general we might discover quite different pathways to a given final state, and/or more than one possible final state.

The `findPath` query can also be used to check whether a model can generate expected intermediate states. For example, proposition `praf1` expresses the property that a certain collection of bindings occurs.

```
eq PD( out {CM | cm
      e((Raf1,(S 621)), (14-3-3a,SBD))
      e((Raf1,C1), b(PS)) e((Raf1,PABM), b(PA))
      e((14-3-3a,DMD), (14-3-3b,DMD))
      {cyto}} ) |= praf1 = true .
```

Executing the query `findPath(qraf,praf1)` results in a path in which rules 1, 2, and 3 have been applied.

Although these results seem satisfactory, we might be concerned that the rules could also generate impossible or unlikely states, such as one in which Raf1 is bound to both 14-3-3's in the dimer as well as being bound to PS and PA. To determine whether this possibility is predicted by the model, we can search for a cell state satisfying `praf2`, defined by matching the pattern

```
PD( out {CM | cm [H-Ras - GTP]
     e((14-3-3a, DMD), (14-3-3b, DMD))
     e((Raf1,(S 621)), (14-3-3a,SBD))
     e((Raf1,(S 259)), (14-3-3b,SBD))
     e((Raf1,C1), b(PS)) e((Raf1,PABM), b(PA))
     {cyto}} )
```

Indeed executing the query `findPath(qraf,praf2)` Maude confirms that such a state is not reachable by returning the result `(noPath).SimplePath`.

4 Conclusions

Pathway Logic is an example of how logical formalisms and formal modeling techniques can be used to develop a new science of symbolic systems biology. We believe that this computational science will provide researchers with powerful tools to facilitate the understanding of complex biological systems and accelerate the design of experiments to test hypotheses about their functions in vivo. In particular, we are interested in formalizing models that biologists can use to think about signaling pathways and other processes in familiar terms while allowing them to computationally ask questions about possible outcomes. Here we have exemplified our approach using the biochemistry of signaling involving the mammalian Raf1 protein kinase.

The use of a logic such as rewriting logic for this kind of modeling has many practical benefits, including the ability to (1) build and analyze models with multiple levels of detail, (2) represent general rules, (3) define new kinds of data and properties, and (4) execute queries using logical inference.

Model validation is done both by experimental testing of predictions and by using the analysis tools to check consistency with known results. Already the Pathway Logic models are useful for clarifying and organizing experimental data from the literature. The eventual goal is to reach a level of maturity that supports prediction of new and possibly unexpected results.

Acknowledgments. We thank the anonymous reviewers for their helpful criticisms. This work was supported in part by grant CA73807 from the National Institutes of Health (KL). Maude tool development has been supported by NSF grants CCR-9900326 and CCR-9900334, and DARPA through Air Force Research Laboratory Contract F30602-02-C-0130.

1. S. Eker et al. Pathway logic: Symbolic analysis of biological signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, January 2002.
2. S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott. Pathway logic: Executable models of biological networks. In *Fourth International Workshop on Rewriting Logic and Its Applications (WRLA'2002)*, 2002. `http://www.elsevier.nl/locate/entcs/volume71.html`.
3. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
4. J. M. Kyriakis and J. Avruch. Mammalian mitogen-activated protein kinase signal transduction pathways activated by stress and inflammation. *Physiol. Rev.*, 81:807–869, 2001.
5. G. Pearson et al. Mitogen-activated protein (MAP) kinase pathways: regulation and physiological functions. *Endocr. Rev.*, pages 153–183, 2001.
6. J.D. Jordan, E. Landau, and R. Iyengar. Signaling networks: The origins of cellular multitasking. *Cell*, 103:193–200, 2000.

7. J. L. Peterson. *Petri Nets: Properties, analysis, and applications*. Prentice-Hall, 1981.
8. P. J. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology using stochastic Petri nets. *Proc. Natl. Acad. Sci. U. S. A.*, 95:6750–6755, 1998.
9. H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid Petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing*, volume 5, pages 341–352, 2000.
10. H. Genrich, R. Küffner, and K. Voss. Executable Petri net models for the analysis of metabolic pathways. *Int. J. STTT*, 3, 2001.
11. J. S. Oliveira et al. A computational model for the identification of biochemical pathways in the Krebs cycle. *J. Computational Biology*, 10:57–82, 2003.
12. I. Zevedei-Oancea and S. Schuster. Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology*, 3(0029), 2003.
13. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
14. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 6, pages 459–470. World Scientific Press, 2001.
15. C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 2001. in press.
16. D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.
17. N. Kam, I.R. Cohen, and D. Harel. The immune system as a reactive system: Modeling t cell activation with statecharts. *Bulletin of Mathematical Biology*, 2002. to appear.
18. S. Efroni, D. Harel, and I.R. Cohen. Towards rigorous comprehension of biological complexity: Modeling, execution and visualization of thymic t-cell maturation. *Genome Research*, 2003. Special issue on Systems Biology, in press.
19. W. Damm and D. Harel. Breathing life into message sequence charts. *Formal Methods in System Design*, 19(1), 2001.
20. N. Kam et al. Formal modeling of C.elegans development: A scenario-based approach. In *First International Workshop on Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 4–20. Springer, 2003.
21. W. Kolch. Meaningful relationships: The regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *Biochem J.*, 351:289–305, 2000.
22. A. S. Dhillon and W. Kolch. Untying the regulation of the Raf-1 kinase. *Arch. Biochem Biophys*, 404:3–9, 2002.