Rewriting Logic and Its Applications in Biology

Part 2: Pathway Logic

Temur Kutsia

RISC, Johannes Kepler University of Linz, Austria
`kutsia@risc.uni-linz.ac.at`

May 28, 2009

Contents

What Is Pathway Logic?

- An approach to modeling biological entities and processes based on rewriting logic.
- An example of how formal modeling techniques can be used to develop a science of symbolic systems biology.
- Uses rewrite theories to formalize the informal models that biologists commonly use to explain biological processes.
- Its formal theories can include both specific facts and general principles relating and categorizing data elements and processes.

Pathway Logic: Motivation

Advantages of representing biological knowledge using formal rules and concepts:

- Data can be interpreted, combined, and queried in the context of biological knowledge.
- Theories concerning different types of information can be combined using well understood operations for combining logical theories.
- A wide range of analytical tools developed for analysis of computer system specifications can be adapted to carry out new kinds of analysis of experimental data curated into formal theories.

Pathway Logic: Motivation

Biological phenomena can be modeled at different levels of abstraction:

- More abstract levels—most crucial, least supported, less details, more scalable.
- Discrete vs. continuous models: The discrete models, when they can be had, scale better and are more robust and predictable than the continuous ones.
- Analogous to different levels of abstraction to model digital systems: More abstract, discrete level of abstraction has great scaling up advantages compared to less abstract, continuous abstraction level provided by differential equations, or even lower quantum electrodynamics levels (cf. digital vs. analog).

Pathway Logic: Motivation

- Biologists reason about the cell at the discrete level.
- However, mostly it consists of semi-formal, informal, and potentially ambiguous notations for things like pathways, cycles, feedback, etc.
- Pathway logic tries to contribute to introducing new computable mathematical models of cell biology that are at a high enough level of abstraction so that
  - they fit biologist's intuitions and informal notations,
  - they make those intuitions mathematically precise,
  - they provide biologists with the predicative power of mathematical models.

Pathway Logic: Motivation

- Pathway Logic is currently being used for the modeling and analysis of signal transduction and metabolic networks in mammalian cells.
- Pathways leading from different initial conditions can be generated automatically from collections of network elements.
- Models are represented using the Maude system.
- Models can be queried and in silico experiments carried out using the execution, search, and model-checking tools of the Maude system.
- In silico experiments can be performed to study the effects of perturbations of these networks.

Pathway Logic: Motivation

Some current capabilities of Pathway Logic:

- Models with different levels of detail and different levels of abstraction.
- Dynamically generated pathways using search and model-checking.
- Transformation to Petri nets for analysis and visualization.
- Roadmap views of dynamically generated pathways.

Pathway Logic in Action: The Signaling Path Model BP

- Example: Modeling a major receptor-mediated pathway in mammalian cells, focusing on the part centered around the epidermal growth factor receptor (EGFR).
- The model is called BP (BioPathways).
- BP includes more than 650 proteins and more than 500 rules describing possible reaction steps.

Pathway Logic in Action: The Signaling Path Model BP

Overview:

- Representation of relevant biological concepts as a membership equational theory.
- Representation of different types of signaling events using rewrite rules.
- Analysis focusing on two different initial states.

BP: Biological Sorts and Elements

- Two basic sorts: Protein and Chemical.
- Chemical for non-biological entities, e.g. for calcium ion Ca++.
- A sort Thing collects together the various basic sorts from which more complex components are formed.
- The sorts and examples of their specific members:

```
sorts Protein Chemical Thing .
subsorts Protein Chemical < Thing .
ops EGFR EGF Pdk1 PKCe : -> Protein .
ops Ca++ PIP3 : -> Chemical .
```

BP: Protein Modification

- BP contains a comprehensive algebra of protein modification. Only a small part here:

```
sorts Modification ModSet .
subsort Modification < ModSet .
ops GDP GTP act deact : -> Modification .
op none : -> ModSet .
op _ _ : ModSet ModSet -> ModSet [assoc comm id: none] .
op [_-_] : Protein ModSet -> Protein [right id: none] .
```

- ModSet: A sort of multisets of modifications, formed from singletons (expressed by the subsort declaration) by multiset union (represented by juxtaposition).
- none: The empty modification set.
- Sets of modifications are applied to proteins by [_-_], e.g. [EGFR - act]: activated form of EGFR.

BP: Protein Association

- Signaling proteins associate to form functional complexes. Representation:

  ```
  sort Complex .
  subsort Complex < Thing .
  op _:_ : Thing Thing -> Complex [comm] .
  ```

- Two things are associated with : to obtain a multicomponent complex. : is commutative but not associative.
- Example: Inhibitory complex
  (IqGap1 : (Ecadherin : bCatenin))

BP: Protein Compartmentalization

- In cells with nucleus (eukaryotic cells) proteins and other molecules exist in complex mixtures (here called Soup) that are compartmentalized :

```
sorts Soup Enclosure MemType .
subsort Thing < Soup .
op empty : -> Soup .
op _ _ : Soup Soup -> Soup [assoc comm id: empty] .
ops CM NM : -> MemType .
op {_ | _{_}} : MemType Soup Soup -> Enclosure .
```

- The terms of the sort Enclosure have a membrane part and an interior, both with its own constituent soup.
- MemType specifies a particular membrane type: the cell membrane CM or the nucleus membrane NM.
- Soup is a multiset of things. Multiset union of soups is again a soup.

BP: Protein Compartmentalization

- Example:

  {CM | cm:Soup PIP3 [Pdk1 - act] {cyto:Soup PKCe}}

- A cell containing
  - the chemical PIP3 and the activated protein [Pdk1 - act] in the cell membrane
  - the protein PKCe in the cytoplasm (cell interior)
  - The variables cm:Soup and cyto:Soup stand for the remaining cell contents.

BP: Protein Compartmentalization

- Sort Dish serves as a top-level soup, and a container for carrying out in silico experiments:

```
sort Dish .
op PD : Soup -> Dish
```

BP: Biochemical Events

Rewrite rules are used to express biochemical events.

Example

- A biochemical event of binding EGF to the EGFR, the first step in the activation of the EGFR signaling pathway:

  Activated Erk1 is translocated to the nucleus where it is functionally sequestered and can regulate the activity of nuclear proteins including transcription factors.

- The same event expressed in rewrite rules:

```
rl[410.Erk1/2.to.nuc]:
  {CM | cm:Soup
          {cyto:Soup [Erk1 - act] {NM | nm:Soup {nuc:Soup}}}}
  =>
  {CM | cm:Soup
          {cyto:Soup {NM | nm:Soup {nuc:Soup [Erk1 - act]}}}} .
rl[438.Erk.act.Elk]:
  [?Erk1/2 - act] Elk1 => [?Erk1/2 - act] [Elk1 - act] .
```

BP: Biochemical Events

Closer look at the rules:

- The first rule:
```
rl[410.Erk1/2.to.nuc]:
   {CM | cm:Soup
            {cyto:Soup [Erk1 - act] {NM | nm:Soup {nuc:Soup}}}}
   =>
   {CM | cm:Soup
            {cyto:Soup {NM | nm:Soup {nuc:Soup [Erk1 - act]}}}}
```
- Describes the translocation of activated Erk1 from the cytoplasm to the nucleus.
- Rule label 410.Erk1/2.to.nuc gives a meaningful description of reaction paths.

BP: Biochemical Events

Closer look at the rules:

- The second rule:

  ```
  rl[438.Erk.act.Elk]:
     [?Erk1/2 - act] Elk1 => [?Erk1/2 - act] [Elk1 - act] .
  ```

- Convention: Symbols beginning with ? are variables with sort named by omitting the ? (declared somewhere in the module).
- Under this convention, ?Erk1/2 is a variable of sort Erk1/2 (Erk1/2 is a subsort of Protein containing proteins Erk1 and Erk2).

BP: Biochemical Events

Example

- Experimental result:

  In the presence of PIP3, activated Pdk1 recruits PKCe from the cytoplasm to the cell membrane and activates it.

- Expressed in a rewrite rule:

```
rl[757.PIP3.Pdk1.act.PKCe]:
  {CM | cm:Soup PIP3 [Pdk1 - act] {cyto:Soup PKCe}}
  =>
  {CM| cm:Soup PIP3 [Pdk1 - act][PKCe - act] {cyto:Soup}}
  [metadata "cite = 21961415"] .
```

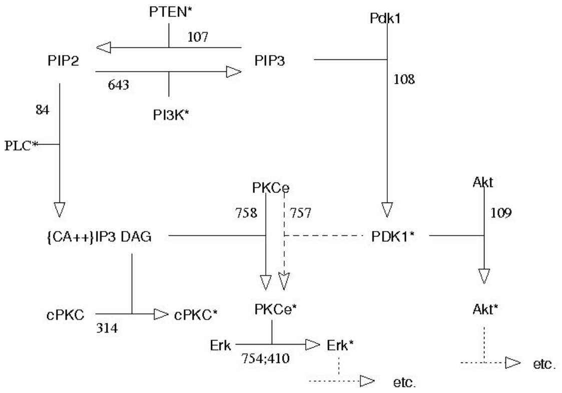
- The metadata attribute
  - allows rules to be annotated with arbitrary information,
  - ignored by the core rewriting engine,
  - available for use by metalevel operations.

Computing with BP

Two examples:

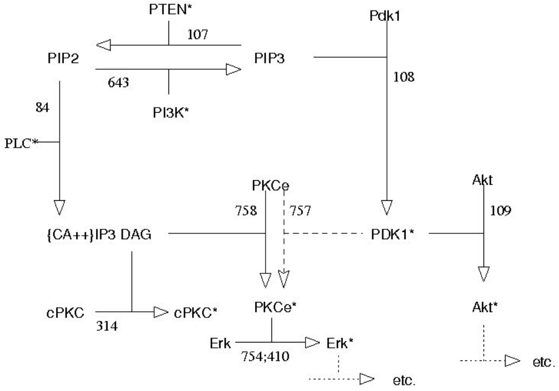
- PKC analysis.
- The EGF-EGFR network.

PKC analysis



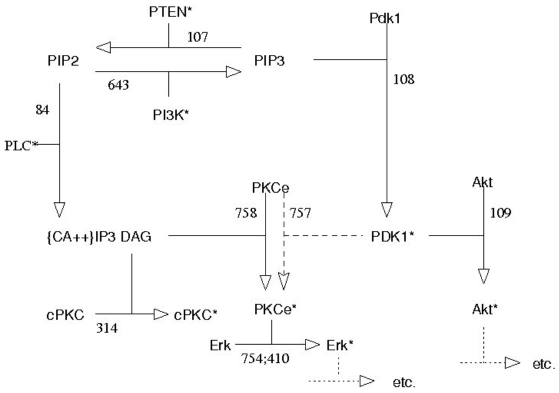
Part of the PKC regulation network. A "*" following a protein indicates that it is activated.

PKC analysis



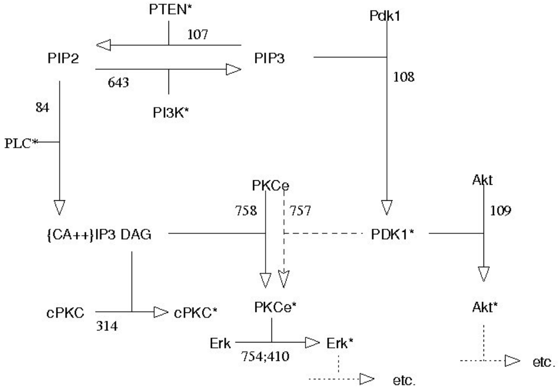
Arrows connect before and after states of reactants.
Numbers denote rules.

PKC analysis



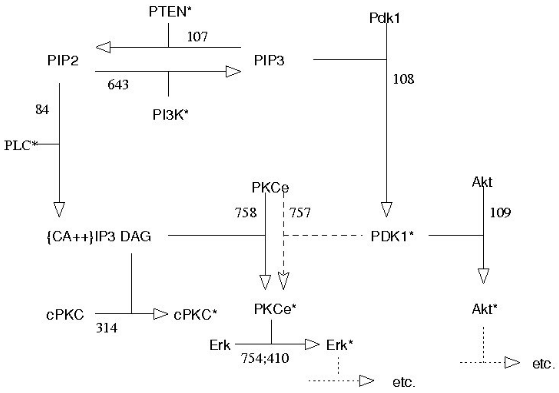
Rule enabling elements that are not consumed are indicated by lines perpendicular to the arrow.

PKC analysis



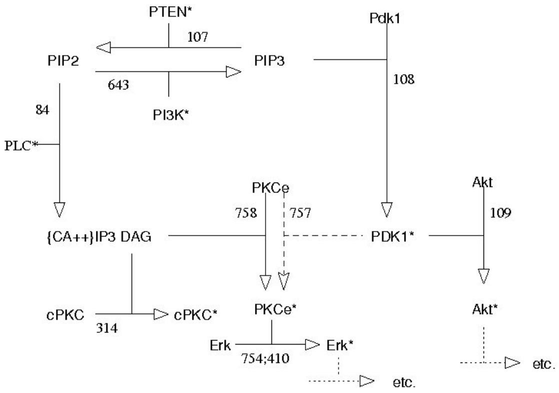
There is an equilibrium between PIP2 and PIP3 (indicated by the pair of arrows between them).

PKC analysis



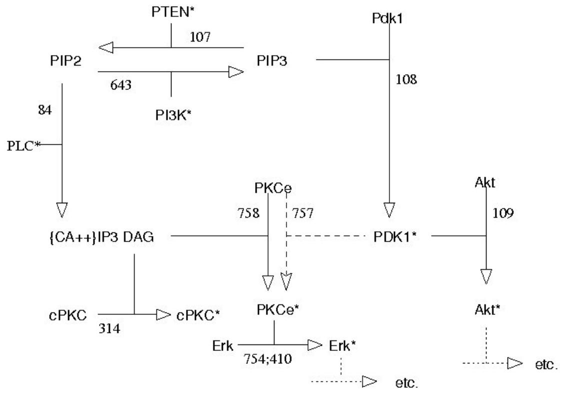
There are (at least) two ways for PKCe to be activated (rules 757 and 758) .

PKC analysis



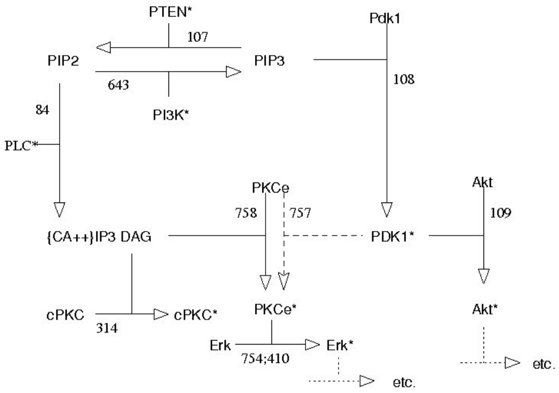
rl [643.PI3KI.mets.PIP2.to.PIP3]:
  PIP2 [?PI3KI - act] => PIP3 [?PI3KI - act] .

PKC analysis



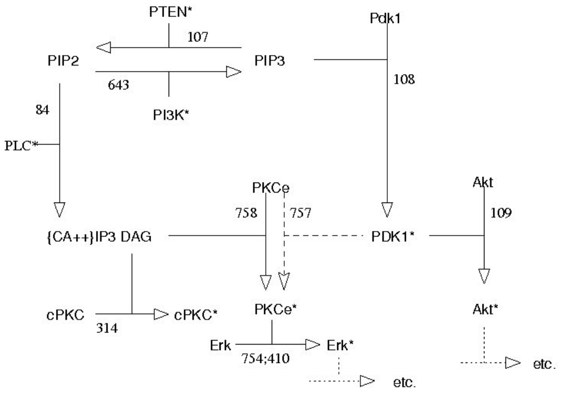
rl [107.Pten.pp.PIP3.to.PIP2]:
    PIP3 [Pten - act] => PIP2 [Pten - act] .

PKC analysis



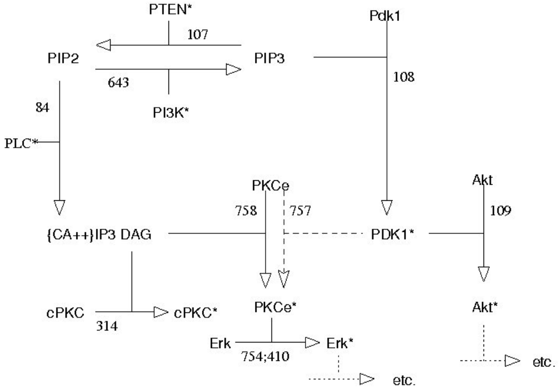
rl [84.PLC.act.PIP2.Ca]:
  Ca++ {CM | cm:Soup PIP2 [?PLC:PLC - act] {cyto:Soup}}
  =>
  {CM | cm:Soup Ca++ DAG IP3 [?PLC:PLC - act] {cyto:Soup}} .

PKC analysis



```
rl [758.DAG.act.nPKC]:
 {CM | DAG cm:Soup {cyto:Soup ?nPKC}}
 =>
 {CM | DAG cm:Soup {cyto:Soup [?nPKC - act]}} .
```

PKC analysis



And so on. We can not show all the rules here.

PKC Analysis

- Analyze the initial state q14 defined as

```
op q14 : -> Dish .
eq q14 = PD(Ca++ {CM | PIP2 [PI3Ka - act]
      [PLCb1 - act] [Pten - act] {Erk1 Pdk1 PKCa PKCe}})
```

- It can be rewritten in various ways.
- Top-down rule-fair rewriting gives

```
rewrite q14 .
result Dish: PD({CM | Ca++ DAG IP3 [PI3Ka - act]
      [PLCb1 - act] [Pten - act] {Erk1 Pdk1 [PKCa - act]
      [PKCe - act] { NM | empty {empty}}}})
```

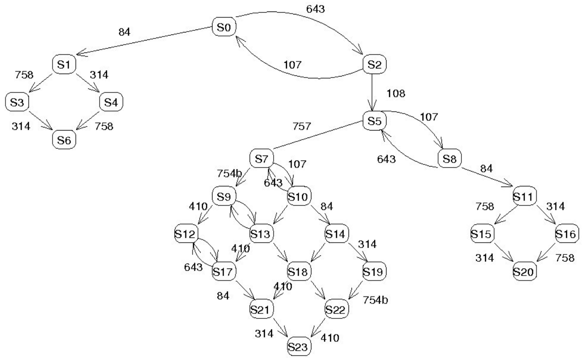
- Position-fair rewriting gives

```
frewrite q14 .
result Dish: PD({CM | Ca++ DAG IP3 [PI3Ka - act]
      [PLCb1 - act] [Pten - act] [Pdk1 - act] {Erk1 [PKCa - act]
      [PKCe - act] {NM | empty {empty}}}})
```

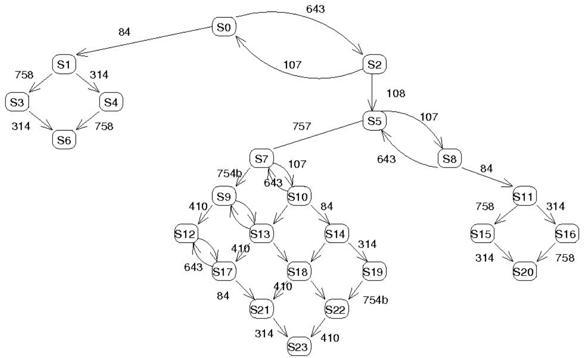
PKC Analysis

- All the possible outcomes starting from q14 can be found using the search command: `search q14 =>! D:Dish .`
- Result: The entire path graph for the PKC regulation network starting with initial state q14.

PKC Analysis

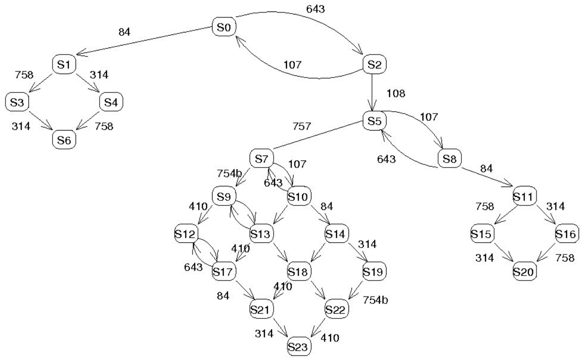


PKC Analysis



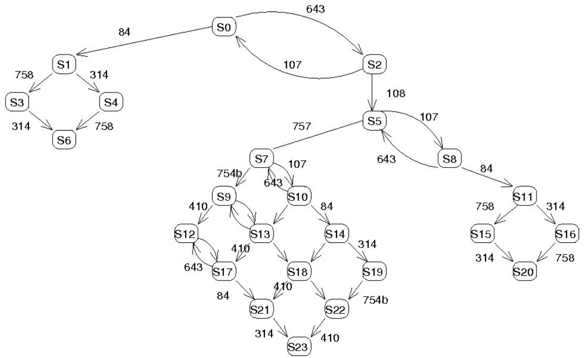
S0 is the starting state q14
S6, S20, and S23 are final states.

PKC Analysis



S6 corresponds to the result obtained by rewrite:
Solution 1 (state 6)
D:Dish –>
PD({CM | Ca++ DAG IP3 [PI3Ka - act]
[PLCb1 - act] [Pten - act] {Erk1 Pdk1 [PKCa - act]
[PKCe - act] {NM | empty {empty}}}})

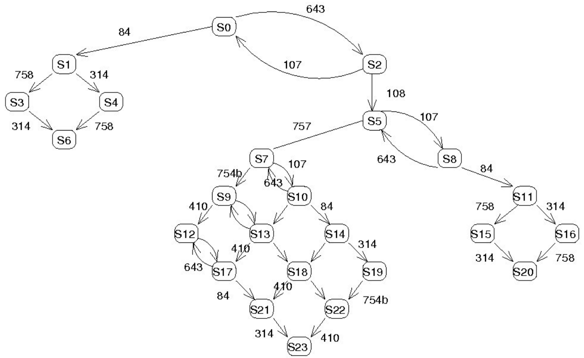
PKC Analysis



S20 corresponds to the result obtained by frewrite:

Solution 2 (state 20)
D:Dish –>
  PD({CM | Ca++ DAG IP3 [PI3Ka - act]
    [PLCb1 - act] [Pten - act] [Pdk1 - act] {Erk1 [PKCa - act]
    [PKCe - act] {NM | empty {empty}}}})

PKC Analysis



In state 23 Erk1 has been activated and translocated from the cytoplasm to the nucleus.:

Solution 3 (state 23)

D:Dish –>

PD({CM | Ca++ DAG IP3 [PI3Ka - act]

[PLCb1 - act] [Pten - act] [Pdk1 - act] [PKCe - act]

{[PKCa - act] {NM | empty {[Erk1 - act]}}}})

The EGF-EGFR network

- Initial state q1:

```
op q1 : -> Dish .
eq q1 = PD(EGF {CM | EGFR Pak1 PIP2 nWasp [H-Ras - GDP]
              {Akt1 Gab1 Grb2 Gsk3 Eps8 Erk1 Mek1 Mekk1
              Mekk4 Mkk4 Mkk3 Mlk2 Jnk1 p38a p70s6k Pdk1
              PI3Ka PKCb1 Raf1 Rsk1 Shc Sos [Cdc42 - GDP]
              {NM | empty {cJun cFos }}}}) .
```

- Two goals:
  1. Find out if a state with cJun and cFos activated is reachable from q1.
  2. Find out if a state with cJun and cFos activated is reachable from q1x (obtained by removing PI3Ka from q1).

The EGF-EGFR network

- Solution idea:
    - Assume that the state specified in the goal is never reachable.
    - Find a counter-example to this assumption by Maude model-checker.

The EGF-EGFR network

Implementation of the solution idea:

- Maude model-checker module defines syntax for LTL (Linear Temporal Logic) formulas built over a sort Prop. It also introduces a sort State and a satisfaction relation $\models$ on states and LTL formulas.
- The user should define corresponding states and propositions in Linear Temporal Logic (LTL) and axiomatize satisfaction on these states and propositions. The LTL semantics lifts satisfaction from propositions to arbitrary LTL formulas.

The EGF-EGFR network

Implementation of the solution idea (cont.):

- In our BP experiments states are dishes, thus Dish is declared to be a subsort of State.
- A proposition prop1 that is satisfied by dishes with cJun and cFos activated is defined as follows:

```
subsort Dish < State .
op prop1 : -> Prop .
eq PD(out:Soup {CM | cm:Soup {cyto:Soup {NM | nm:Soup
            {nuc:Soup [cJun - act] [cFos - act]}}}}) |= prop1
      = true .
```

- The formula ∼ <> prop1 says that prop1 never holds. Executing the command red q1 |= ∼ <> prop1 . results in a counter example - a reachable state satisfying prop1 together with a path leading to that state.
- In the same way, red q1x |= ∼ <> prop1 . results in a counter example.

Topics Not Discussed

Other capabilities of Pathway logic:

- Meta-analysis
- Visualization
- ...

Summary

- Representation of biological signaling networks in Maude.
- Demonstration how such a model can be used for various in silico experiments and advanced forms of symbolic analysis.
- Executable models put new kinds of computational capabilities into the hands of biologists. Their predictive power can be used to generate hypotheses to be tested by other means and to design experiments. They also serve as a starting point for a wide range of exciting applications of formal modeling.

References

Materials used to prepare these lectures:

- Papers on Maude and Rewriting Logic.
  http://maude.cs.uiuc.edu/papers/.
- Pathway Logic web page.
  http://pl.csl.sri.com/.
- M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada.
  A Maude Tutorial.
  SRI International, 2000.

References

S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott.
Pathway Logic: Executable models of biological networks.
ENTCS, 71, 2002.

J. Meseguer.
Bio-Pathway Logic. Slides.

J. Meseguer.
Conditional Rewriting Logic as a unified model of concurrency.
TCS, 96(1):73–155, 1992.

C. Talcott.
Pathway Logic tutorial. Parts 1 and 2. Slides.