Computational Thinking for Life

Jane Hillston.
LFCS, University of Edinburgh

11th January 2006

(Joint work with Muffy Calder, Adam Duguid, Stephen Gilmore, and Marco Stenico)

Systems Biology: Will it Work?

Systems Biology: Will it Work?

- Meeting held in Sheffield, January 2005 under the auspices of the Biochemical Society.

Systems Biology: Will it Work?

- Meeting held in Sheffield, January 2005 under the auspices of the Biochemical Society.
- Varying degrees of optimism with respect to the topic of the workshop.

Systems Biology: Will it Work?

- Meeting held in Sheffield, January 2005 under the auspices of the Biochemical Society.
- Varying degrees of optimism with respect to the topic of the workshop.
- Equally wide spectrum of definitions of what systems biology is and what it is trying to achieve....

Systems Biology: Will it Work?

- Meeting held in Sheffield, January 2005 under the auspices of the Biochemical Society.
- Varying degrees of optimism with respect to the topic of the workshop.
- Equally wide spectrum of definitions of what systems biology is and what it is trying to achieve....
- Perhaps not surprising for a new initiative?

Systems Theory and Biology

- Meeting held in Columbus Ohio, October 1966, organised by Mihajlo Mesarović.

Systems Theory and Biology

- Meeting held in Columbus Ohio, October 1966, organised by Mihajlo Mesarović.
- The third in a series of annual symposia: Systems Approach in Biology.

Systems Theory and Biology

- Meeting held in Columbus Ohio, October 1966, organised by Mihajlo Mesarović.
- The third in a series of annual symposia: Systems Approach in Biology.
- Stated objective — "To assess the past development and the future potential of the application of the systems approach in biology."

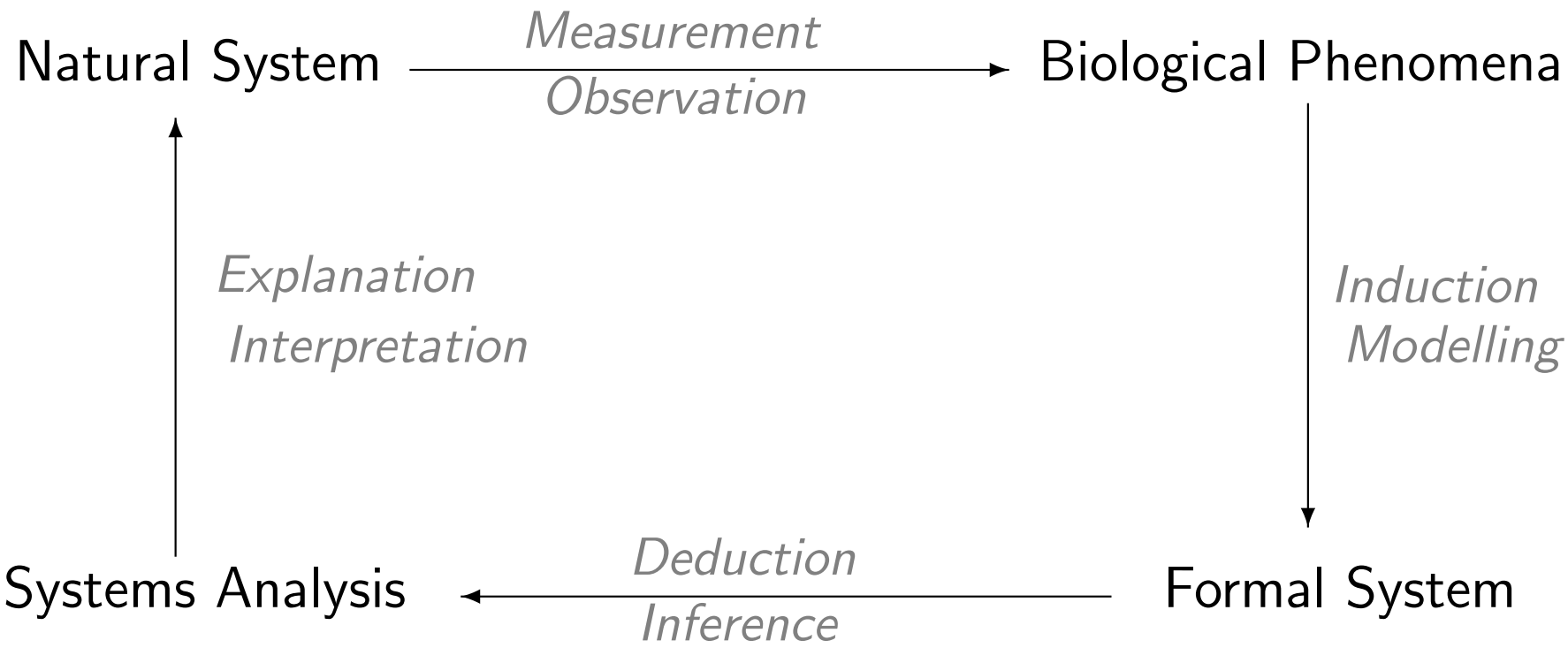
Outline

What is Systems Biology?

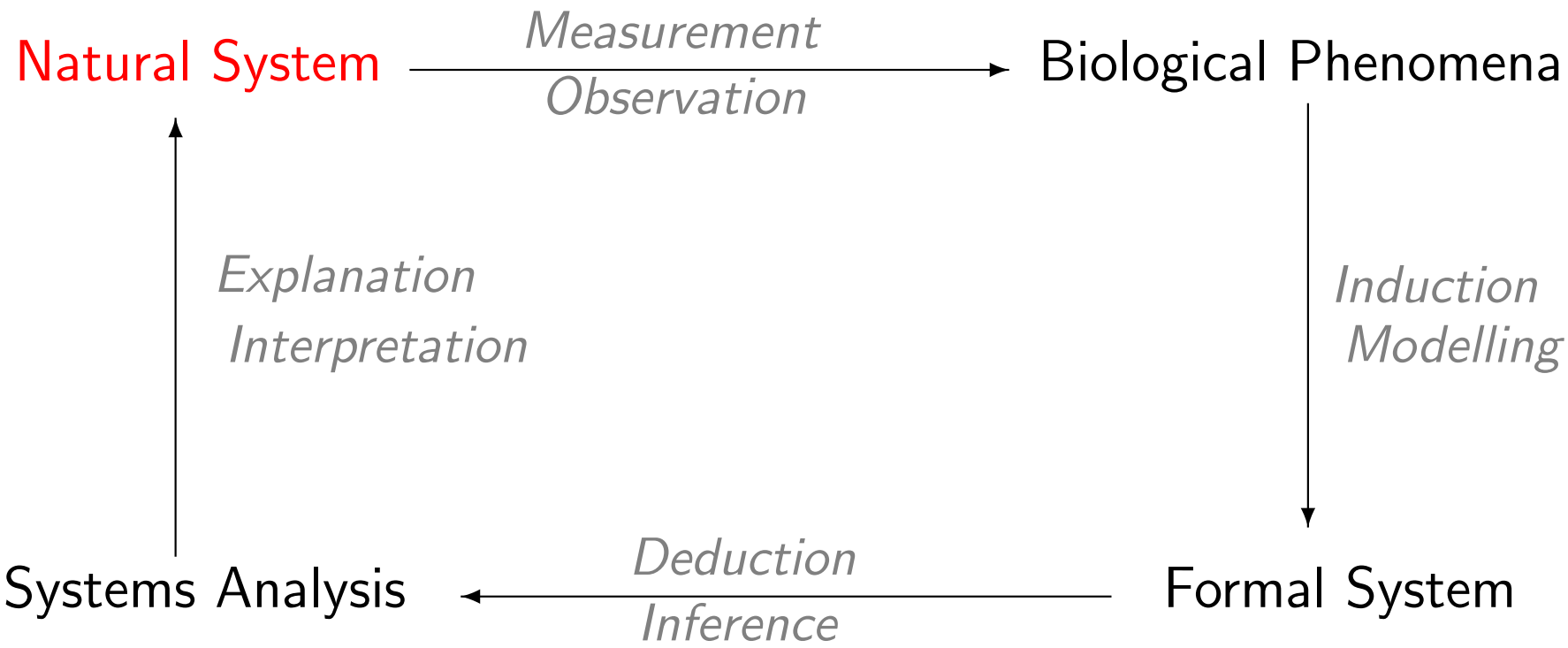
What is Systems Biology?

"The principal aim of systems biology is to provide both a conceptual basis and working methodologies for the scientific explanation of biological phenomena" – Olaf Wolkenhauer

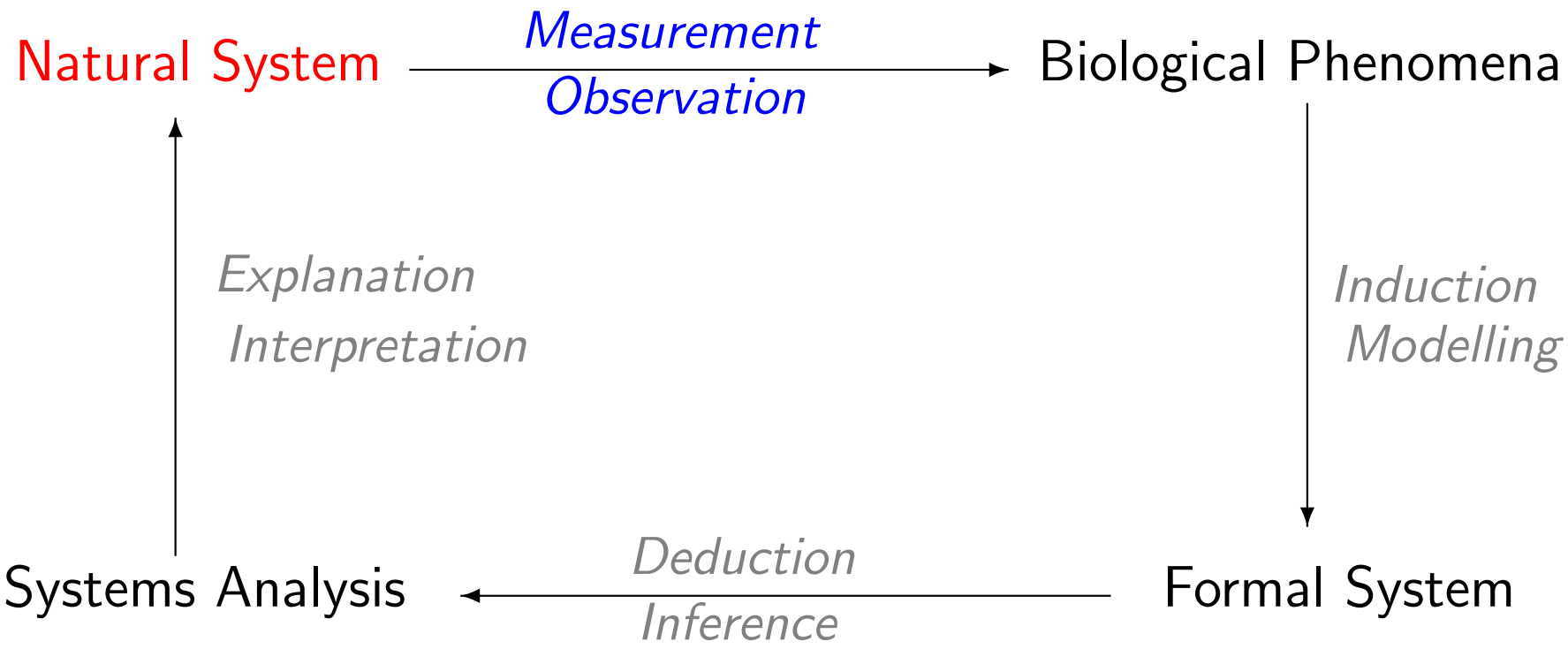
Systems Biology Methodology



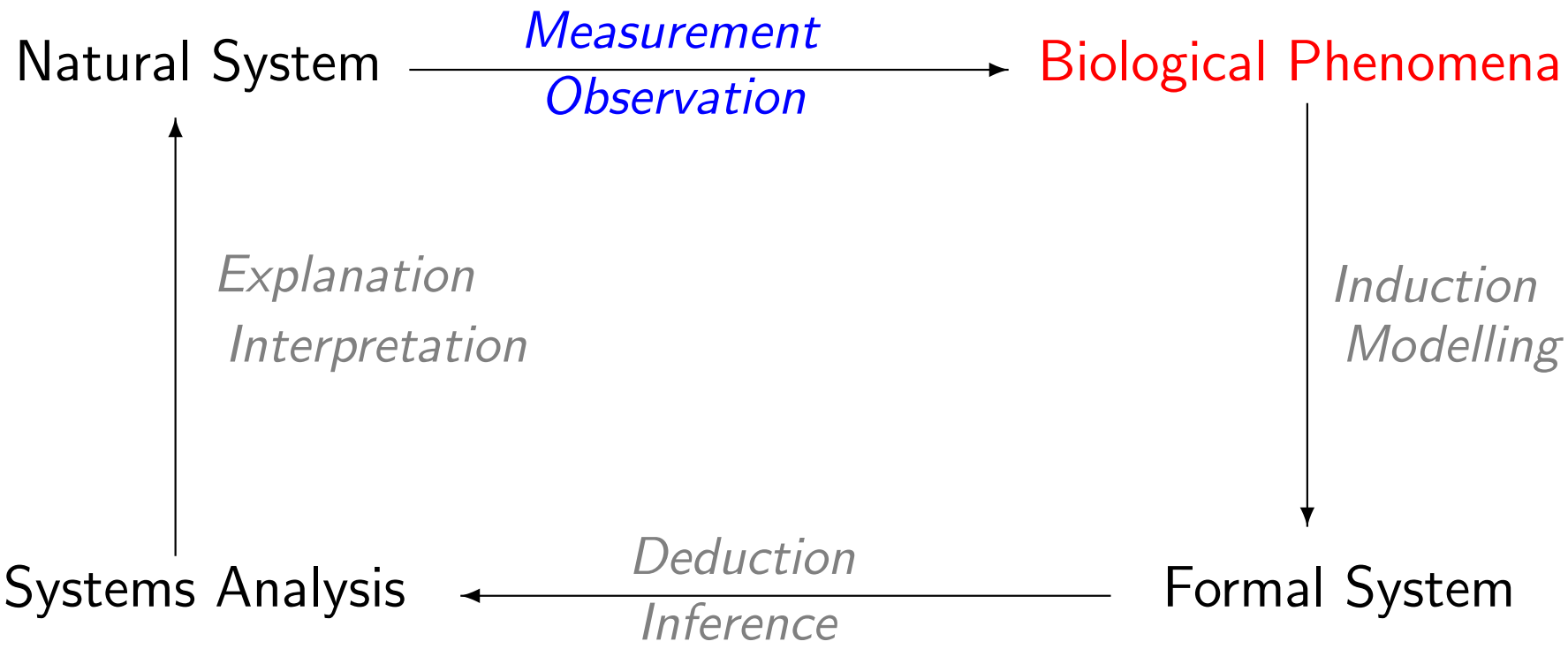
Systems Biology Methodology



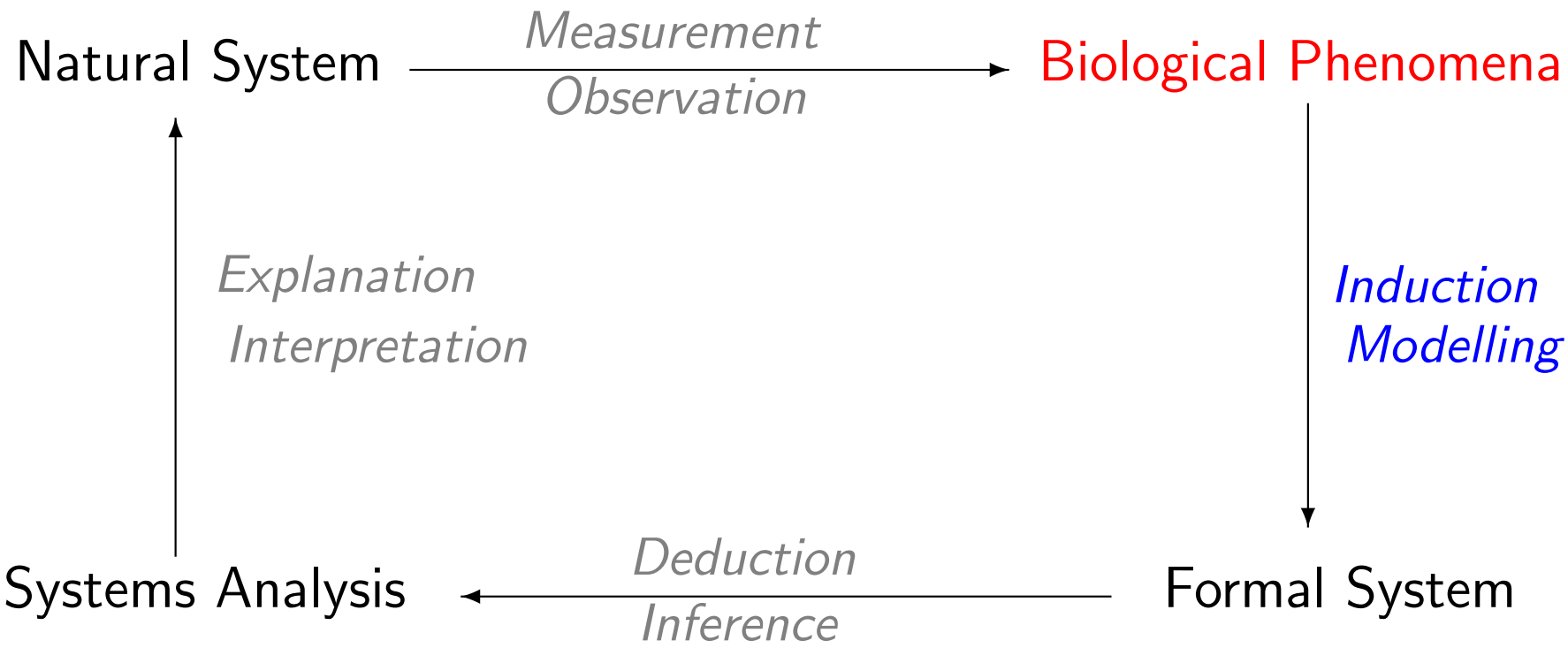
Systems Biology Methodology



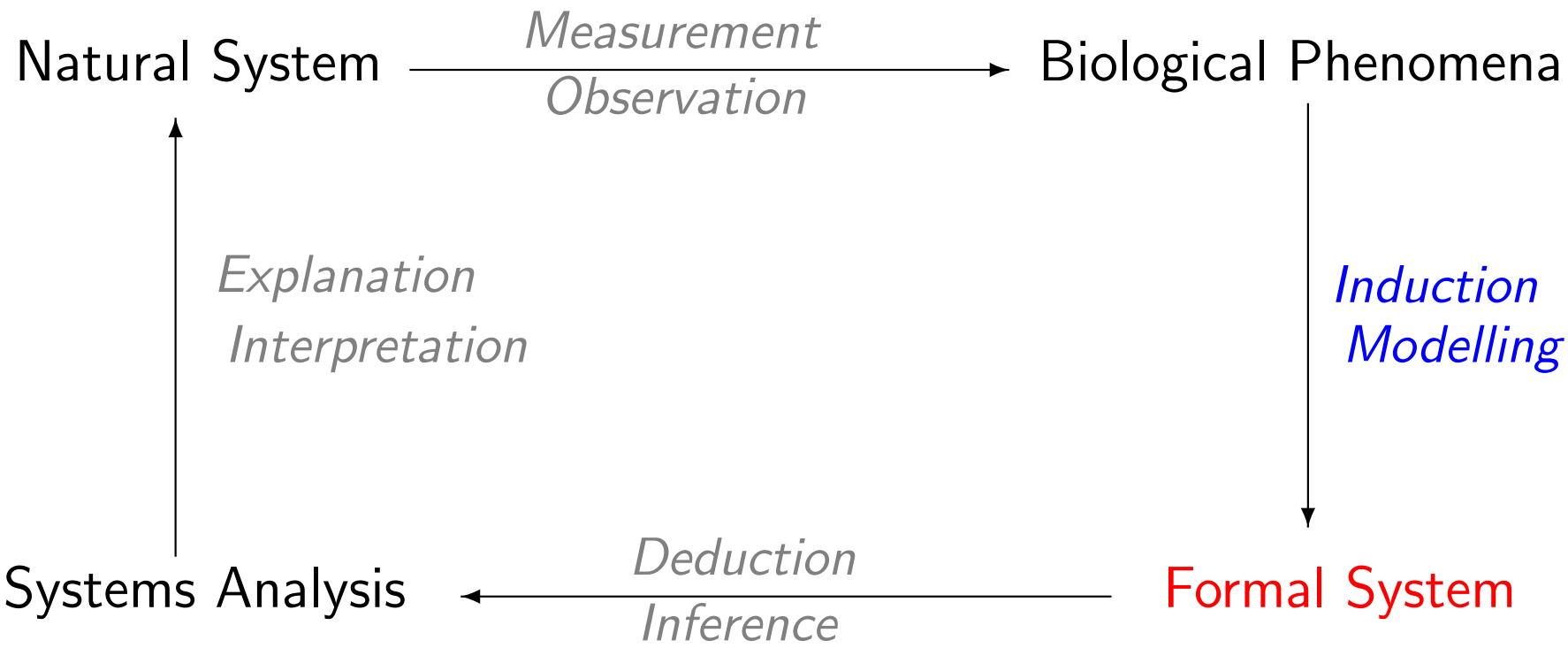
Systems Biology Methodology



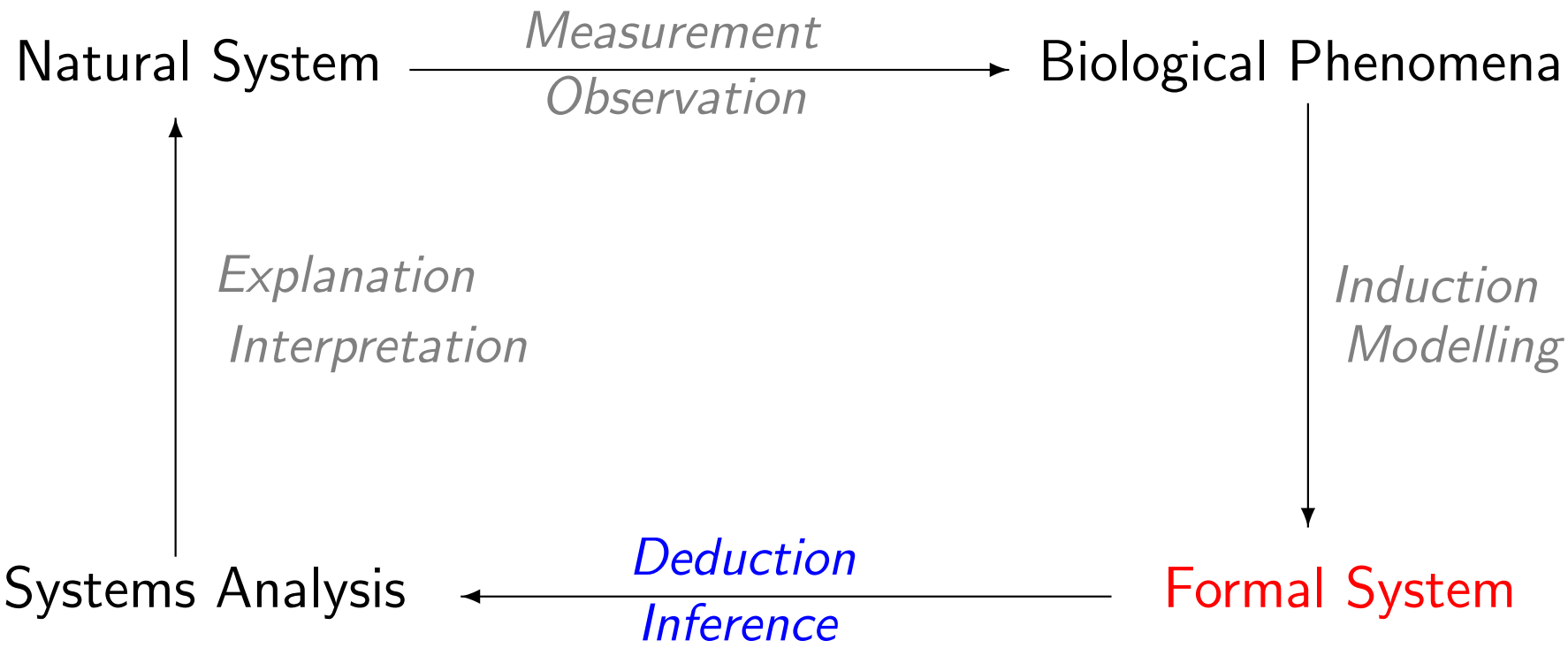
Systems Biology Methodology



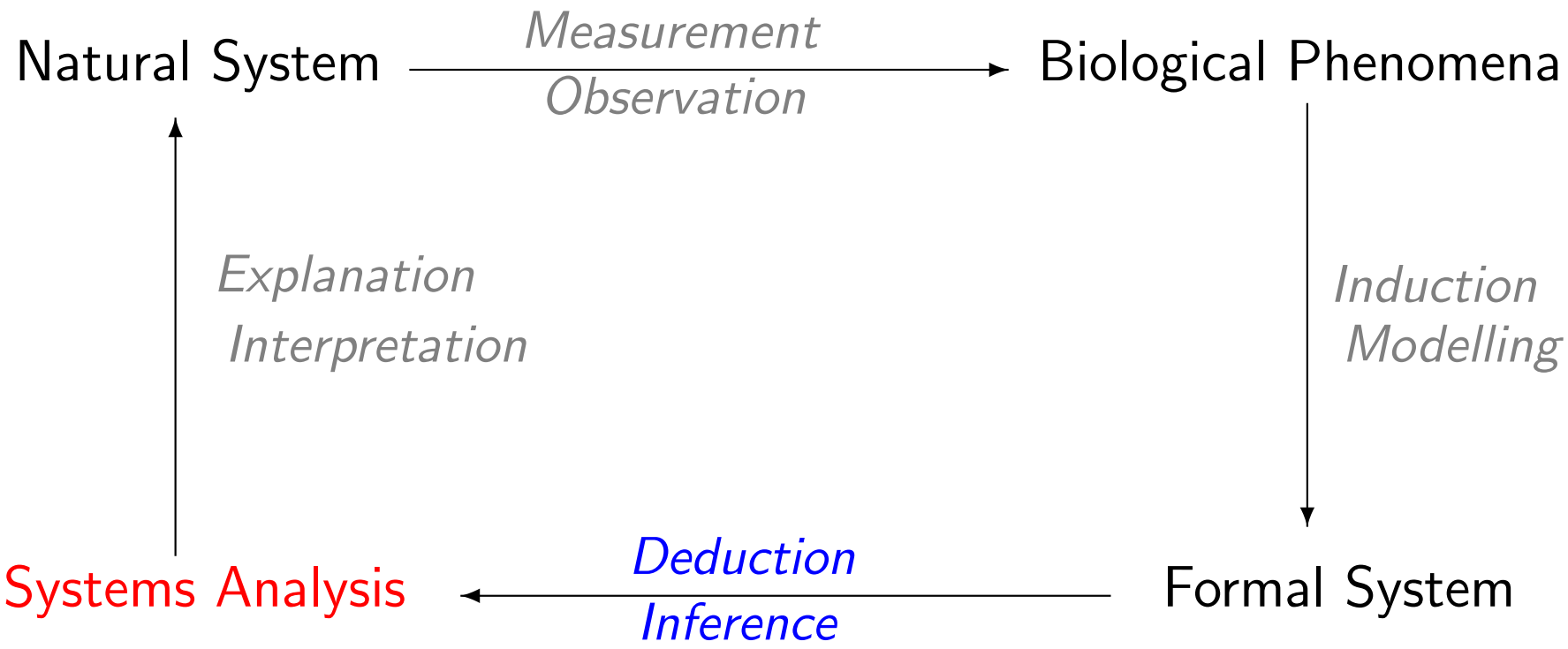
Systems Biology Methodology



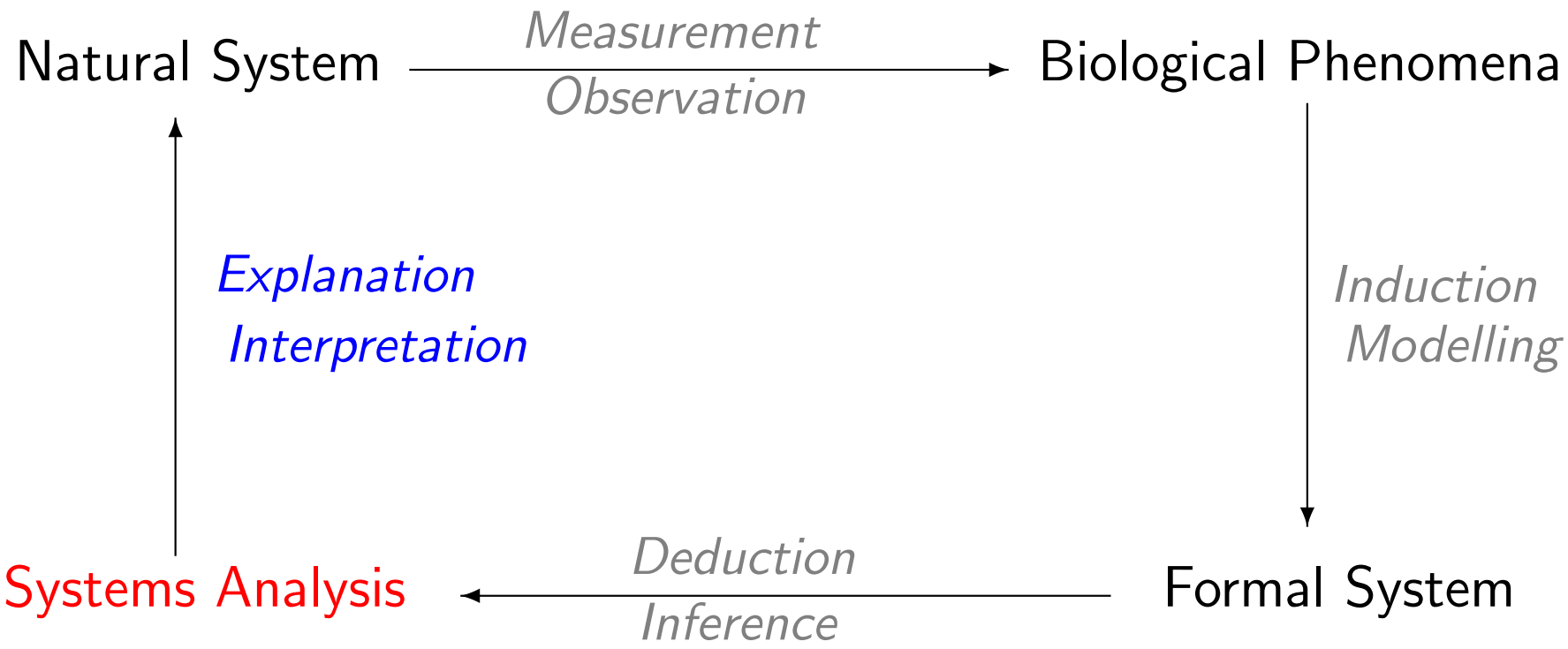
Systems Biology Methodology



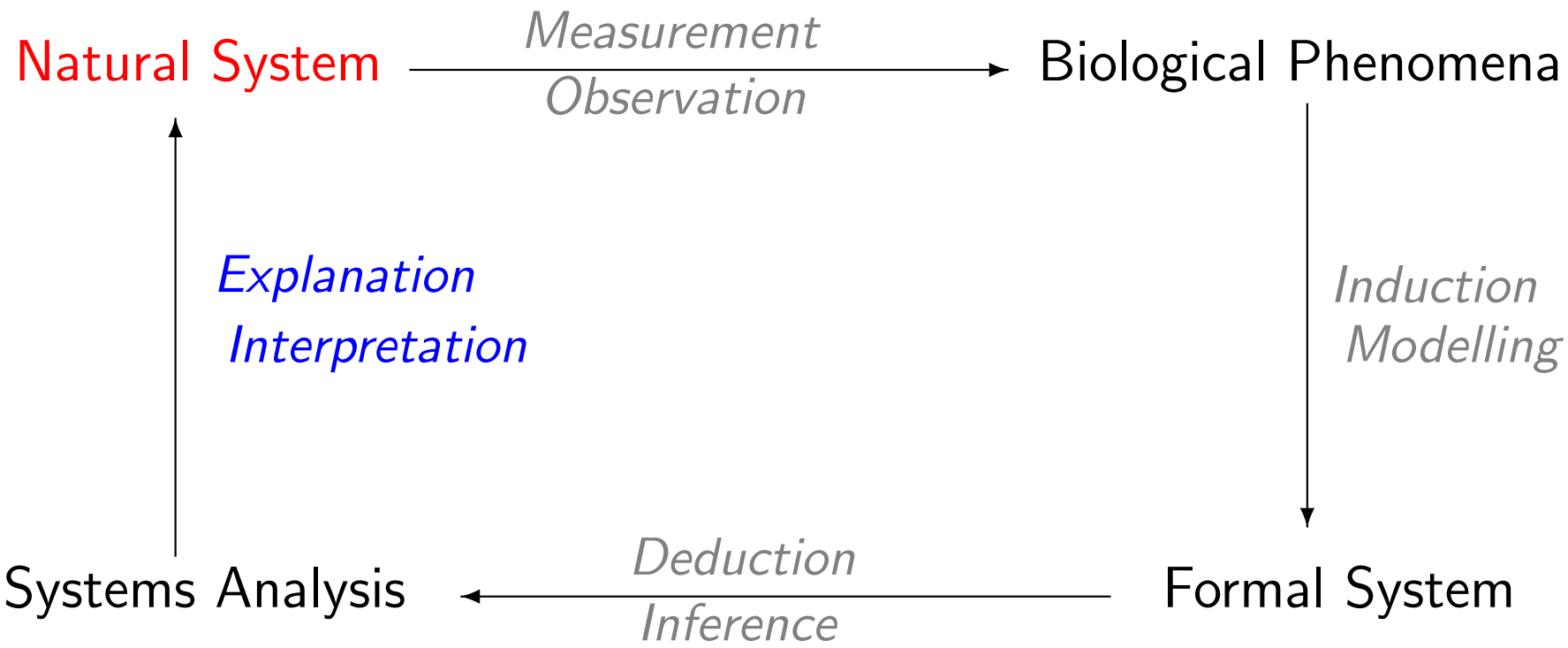
Systems Biology Methodology



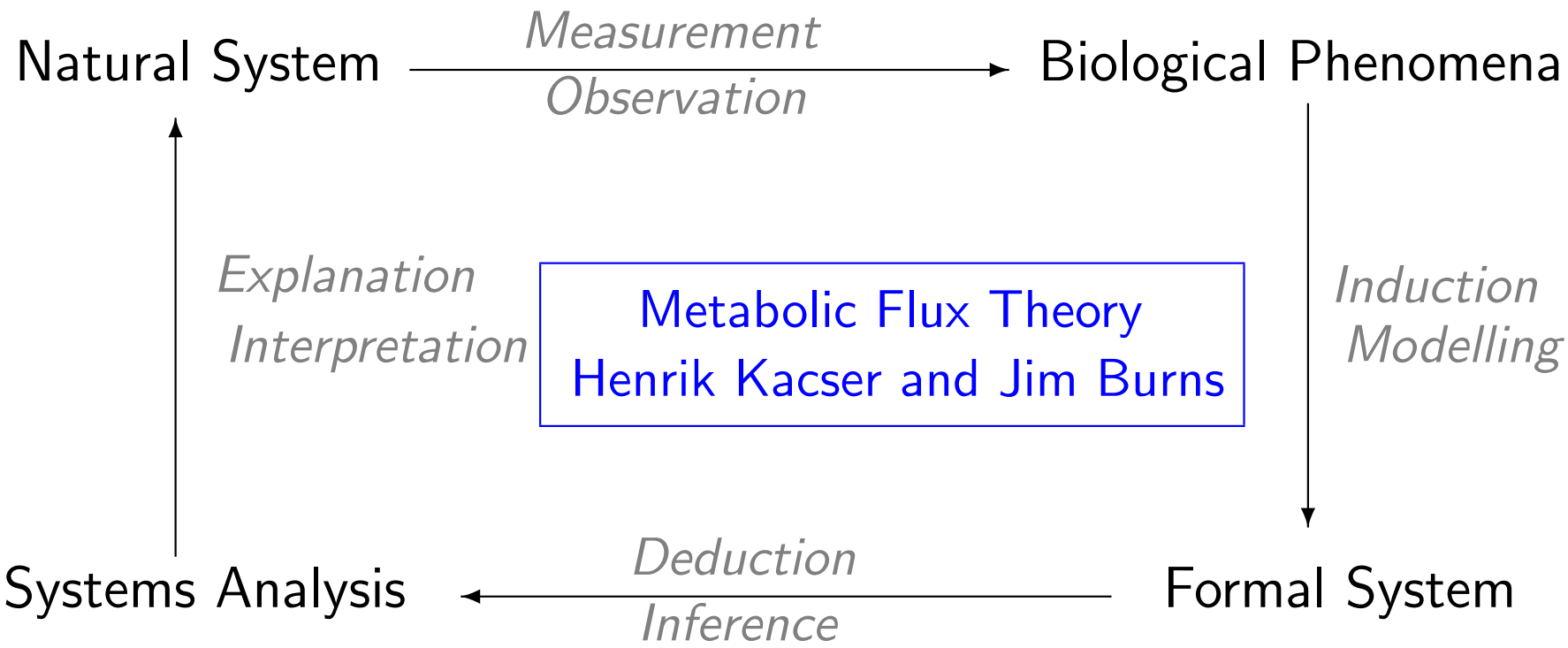
Systems Biology Methodology



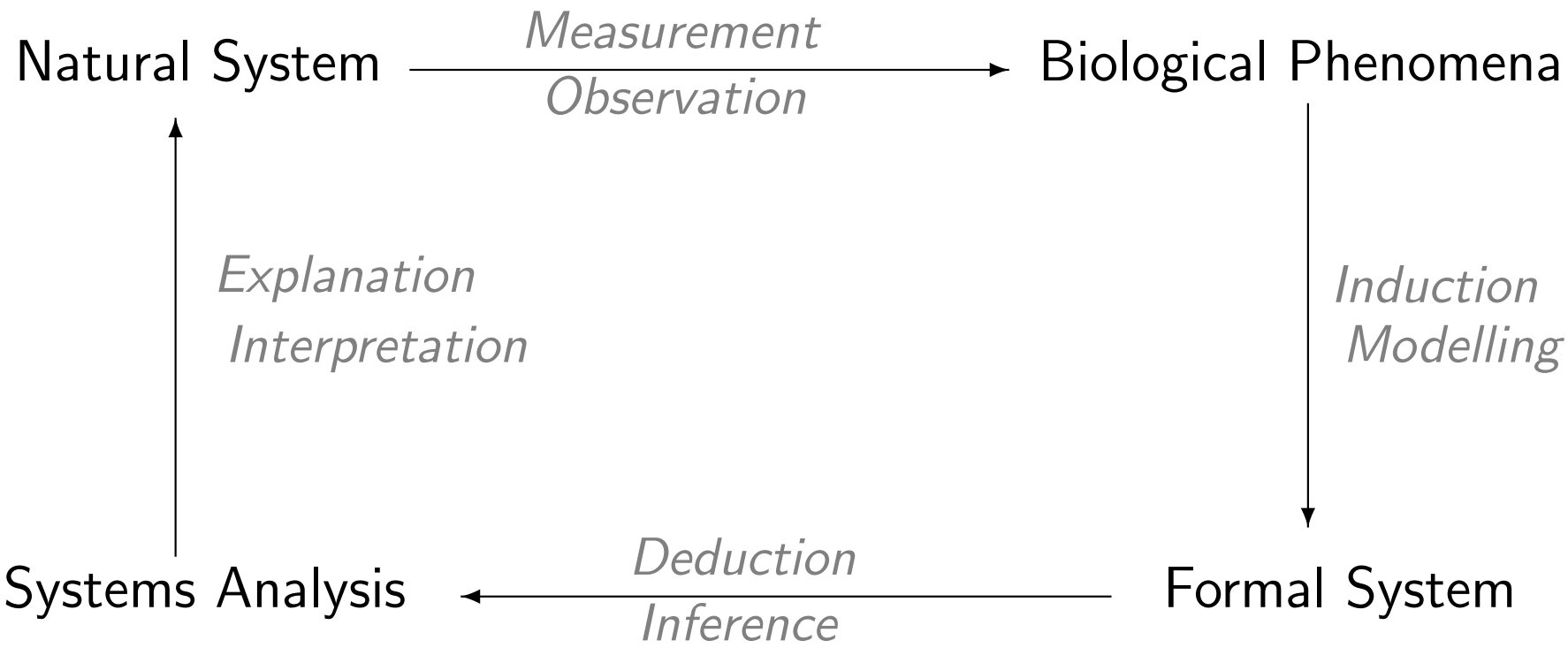
Systems Biology Methodology



Systems Biology Methodology



Systems Biology Methodology



Measurement, Observation and Induction

- Robot Scientist project — Kell, King, Muggleton *et al.*

Measurement, Observation and Induction

- Robot Scientist project — Kell, King, Muggleton *et al.*
- Combination of machine learning for hypothesis generation and genetic algorithms for automatic experimental tuning.

Measurement, Observation and Induction

- Robot Scientist project — Kell, King, Muggleton *et al.*
- Combination of machine learning for hypothesis generation and genetic algorithms for automatic experimental tuning.
- Experiments are carried out by a robot.

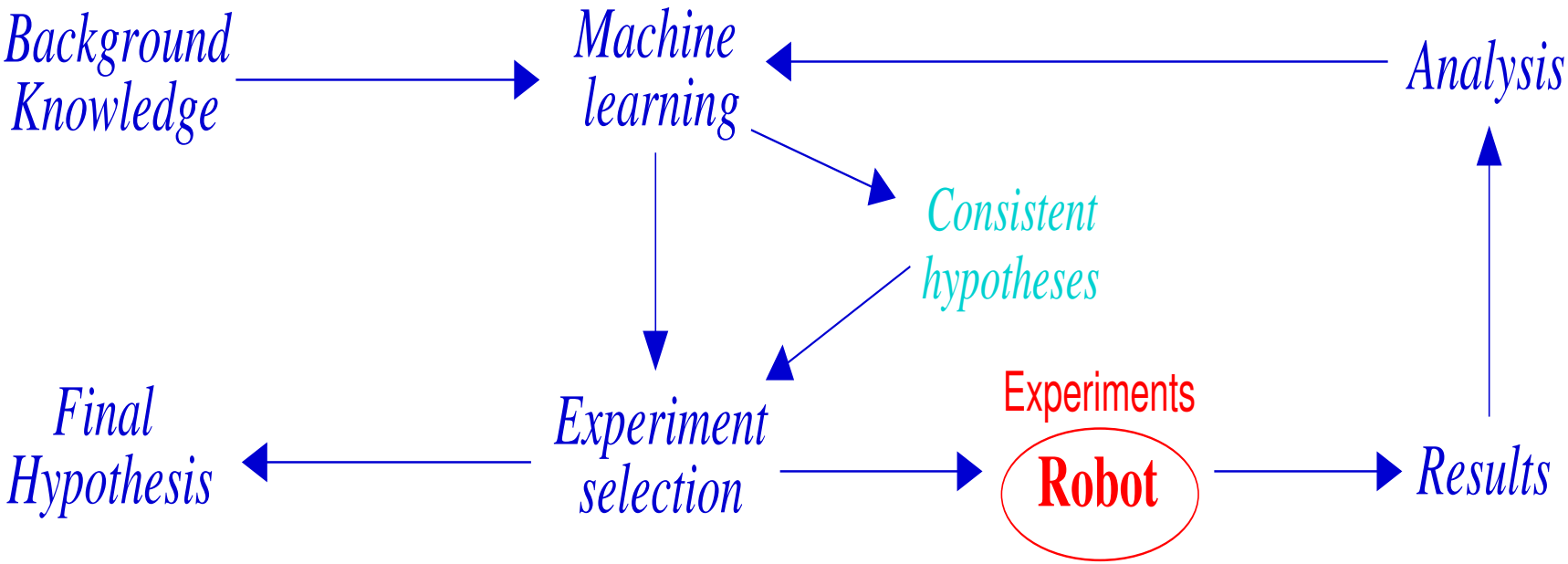
Measurement, Observation and Induction

- Robot Scientist project — Kell, King, Muggleton *et al.*
- Combination of machine learning for hypothesis generation and genetic algorithms for automatic experimental tuning.
- Experiments are carried out by a robot.
- Data is generated at rates which exceed what is possible when there are humans in the loop.

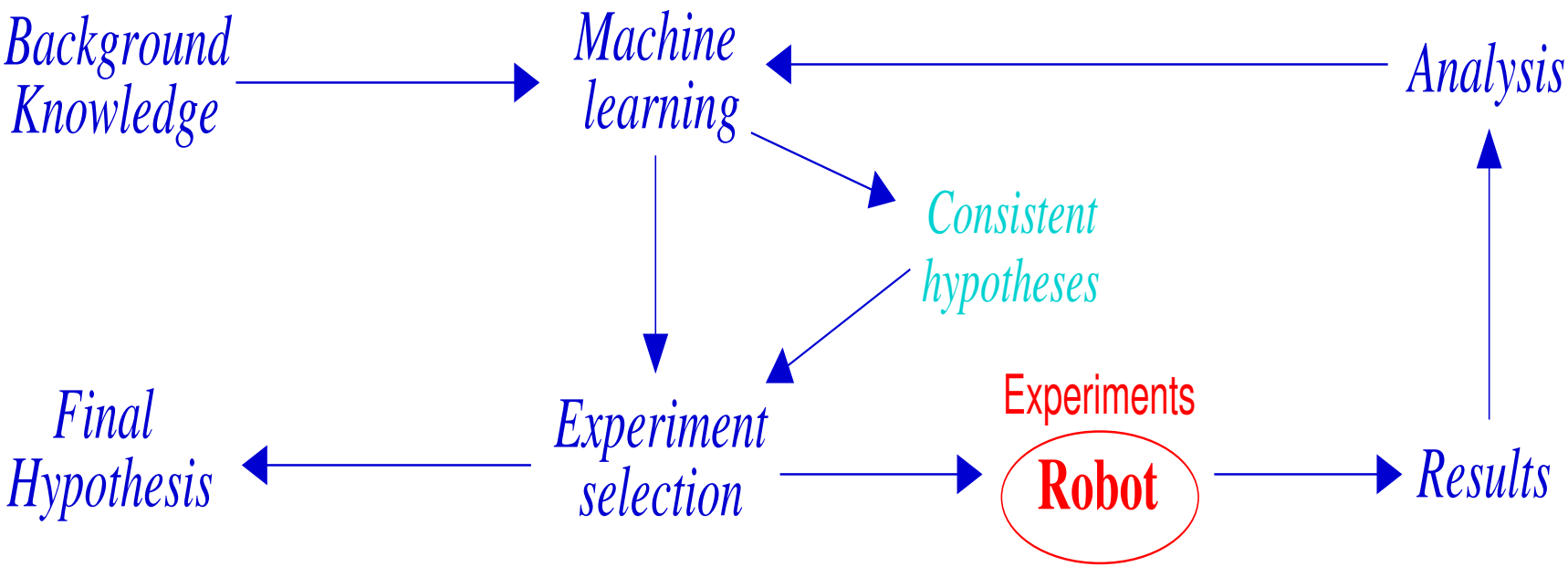
Measurement, Observation and Induction

- Robot Scientist project — Kell, King, Muggleton *et al.*
- Combination of machine learning for hypothesis generation and genetic algorithms for automatic experimental tuning.
- Experiments are carried out by a robot.
- Data is generated at rates which exceed what is possible when there are humans in the loop.
- Moreover the intelligent experiment selection strategy is competitive with (good) human strategies, and significantly outperforms *cheapest* and *random* selection strategies.

The Robot Scientist

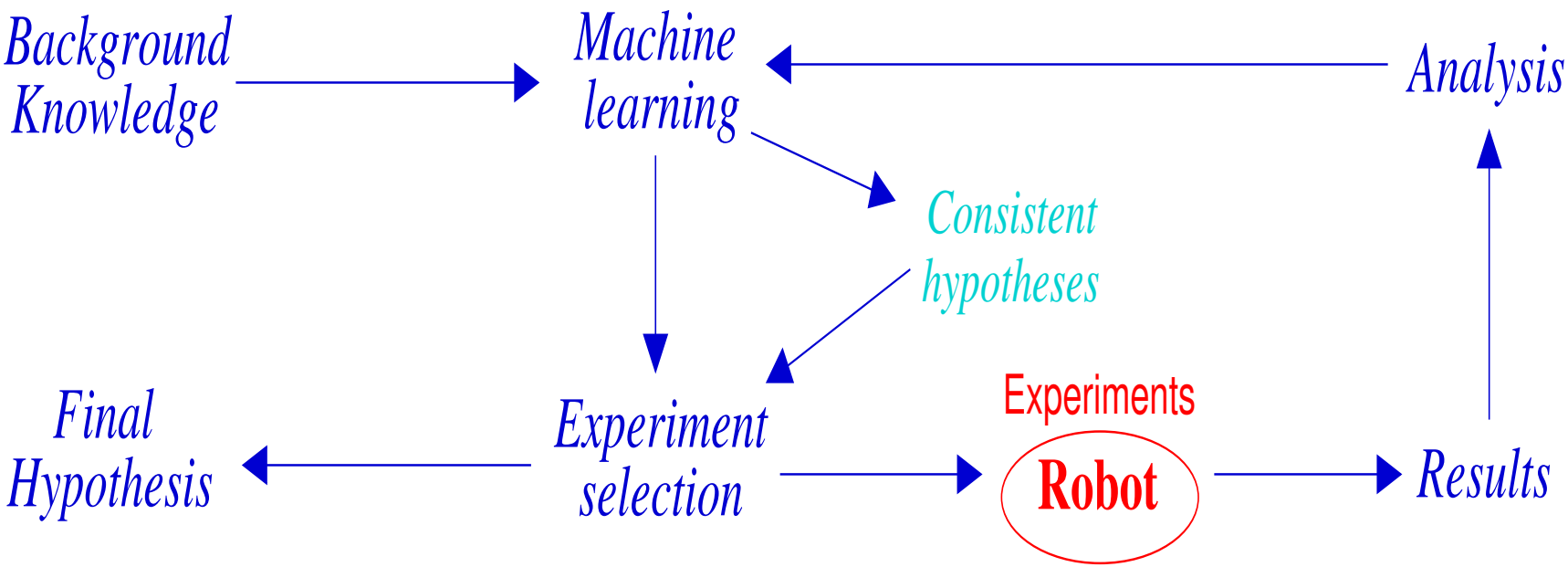


The Robot Scientist



- No human intellectual input in the design of experiments or the interpretation of data.

The Robot Scientist



- No human intellectual input in the design of experiments or the interpretation of data.
- Integrates scientific discovery software with laboratory robotics.

Outline

The Challenges

Systems biology modelling faces a number of challenges. In particular:

- An excess of data, much of which is noisy and/or incomplete;
- The problem of infinite regress;
- Some observations can only be explained by multi-level modelling.

The Challenges

Systems biology modelling faces a number of challenges. In particular:

- An excess of data, much of which is noisy and/or incomplete;
- The problem of infinite regress;
- Some observations can only be explained by multi-level modelling.

The Challenges

Systems biology modelling faces a number of challenges. In particular:

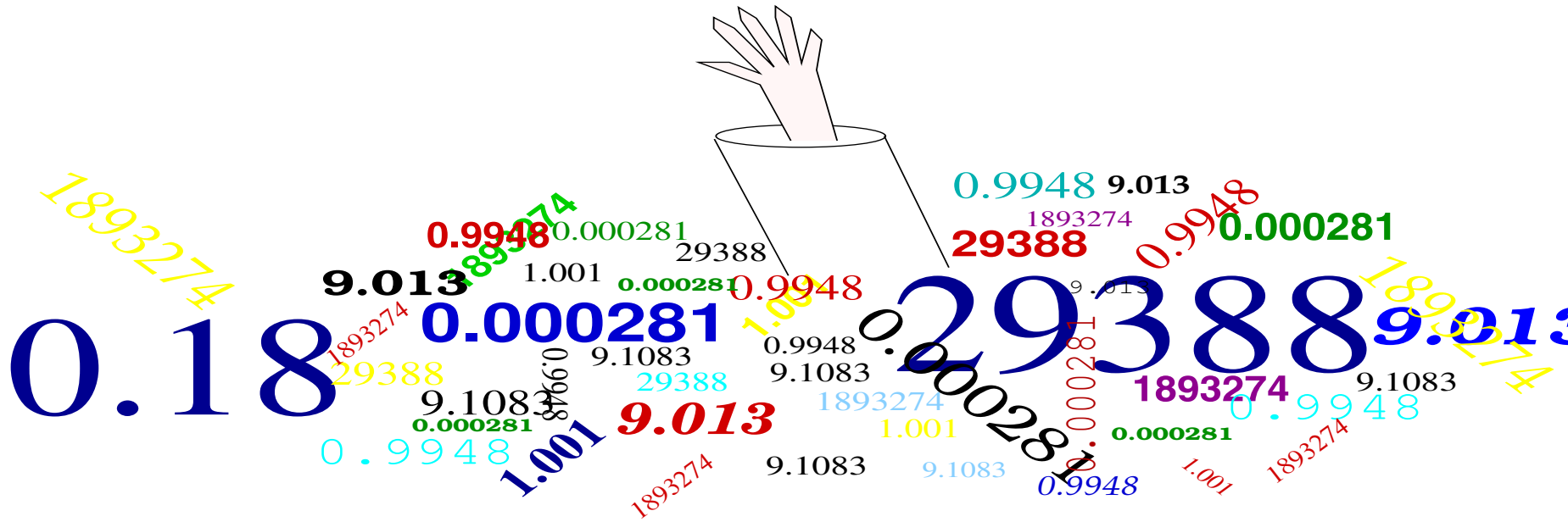
- An excess of data, much of which is noisy and/or incomplete;
- The problem of infinite regress;
- Some observations can only be explained by multi-level modelling.

The Challenges

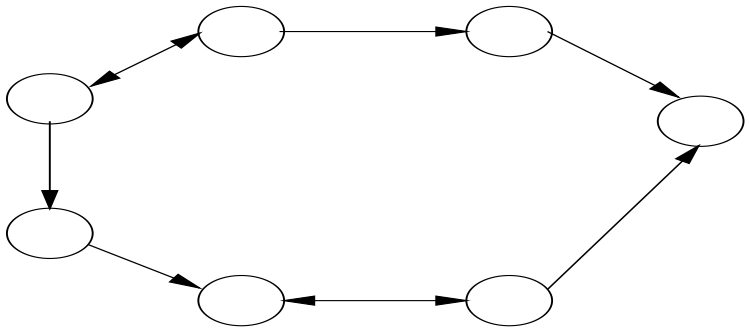
Systems biology modelling faces a number of challenges. In particular:

- An excess of data, much of which is noisy and/or incomplete;
- The problem of infinite regress;
- Some observations can only be explained by multi-level modelling.

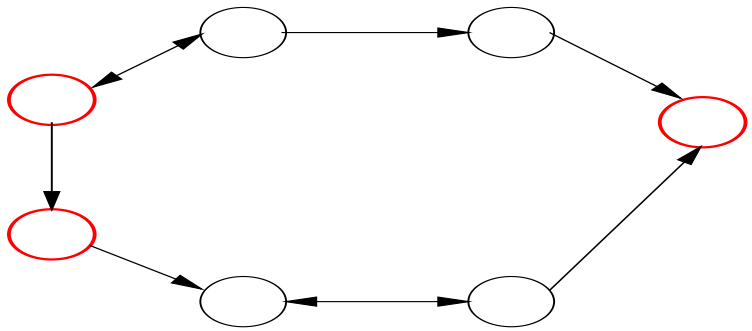
The problem of data



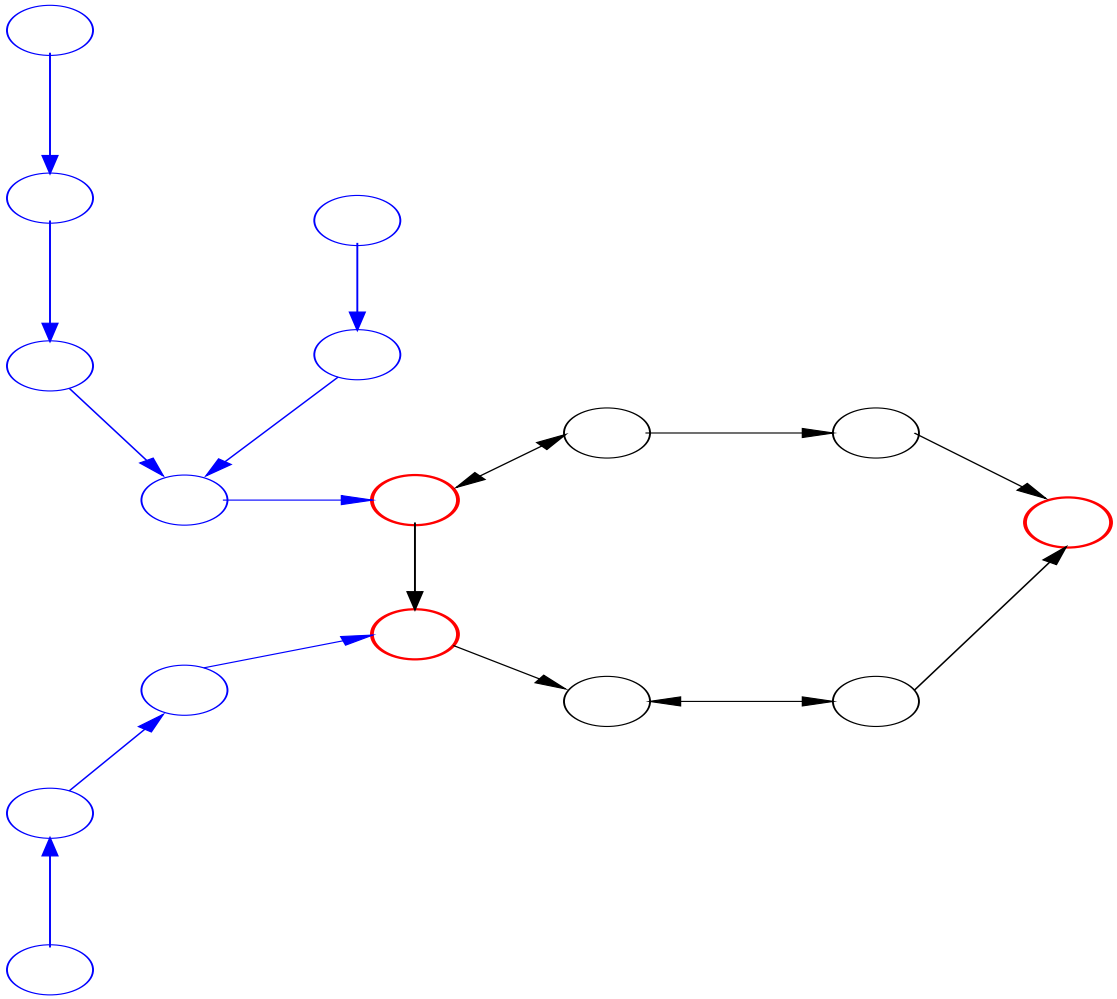
The problem of *Infinite Regress*



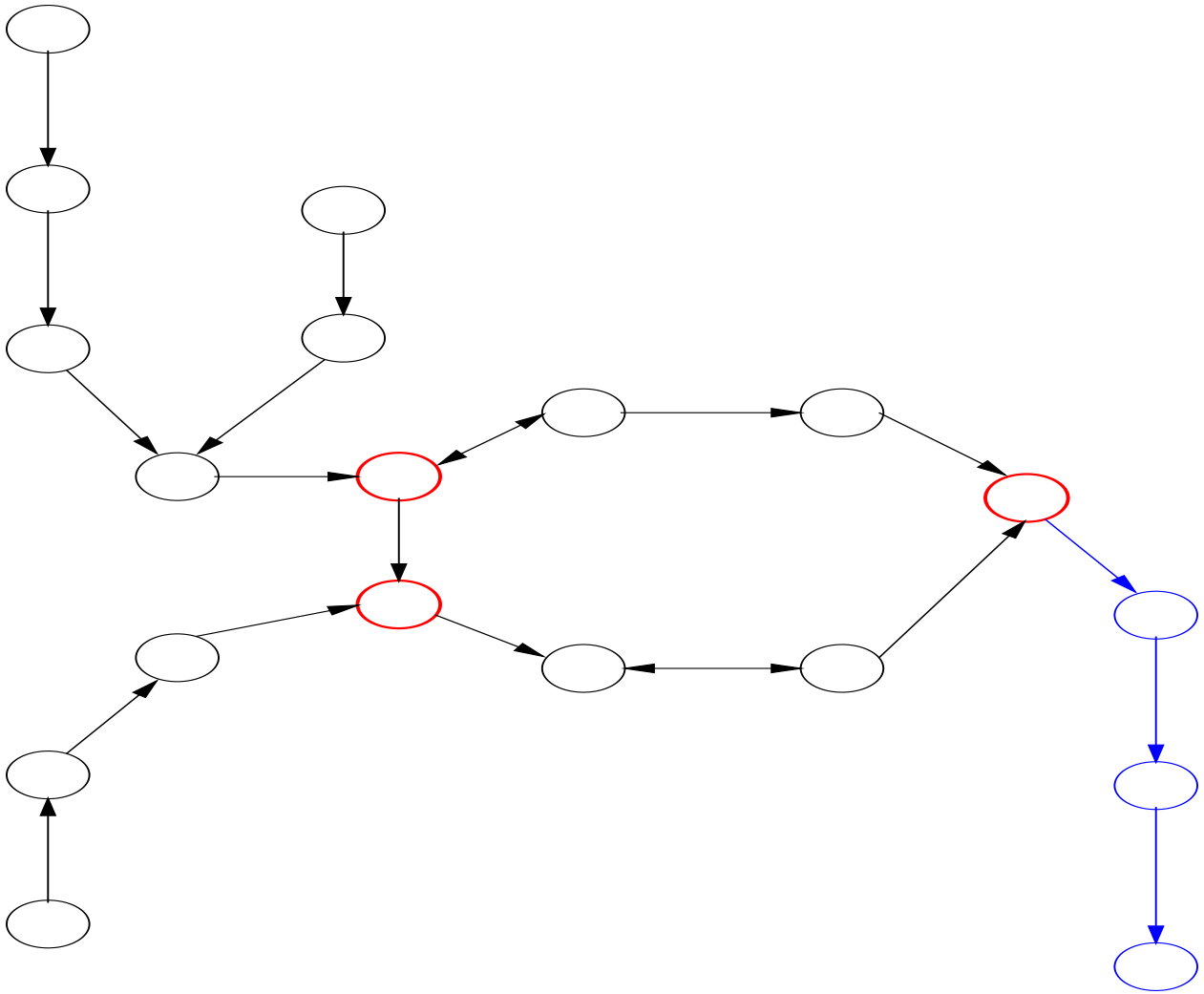
The problem of *Infinite Regress*



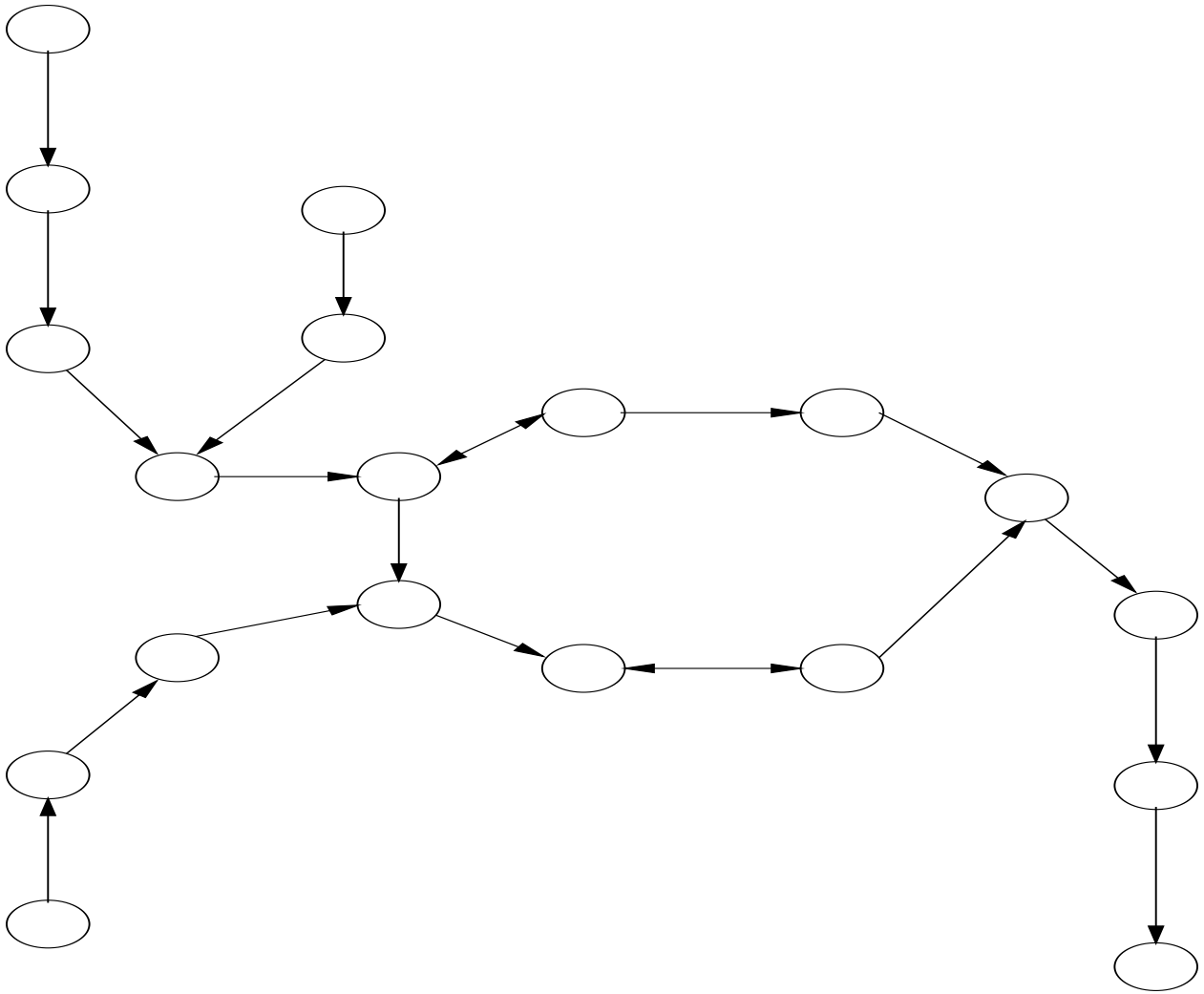
The problem of *Infinite Regress*



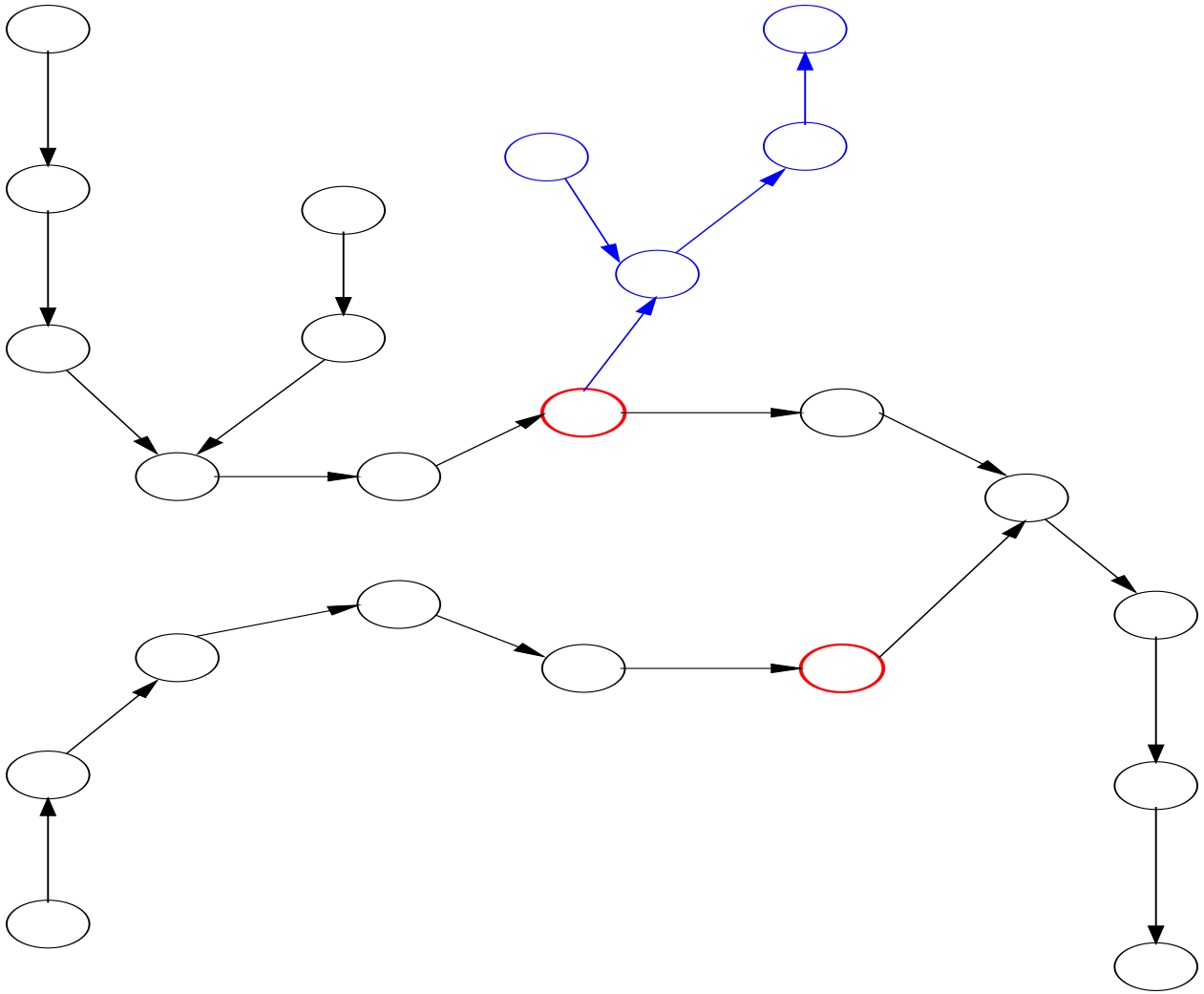
The problem of *Infinite Regress*



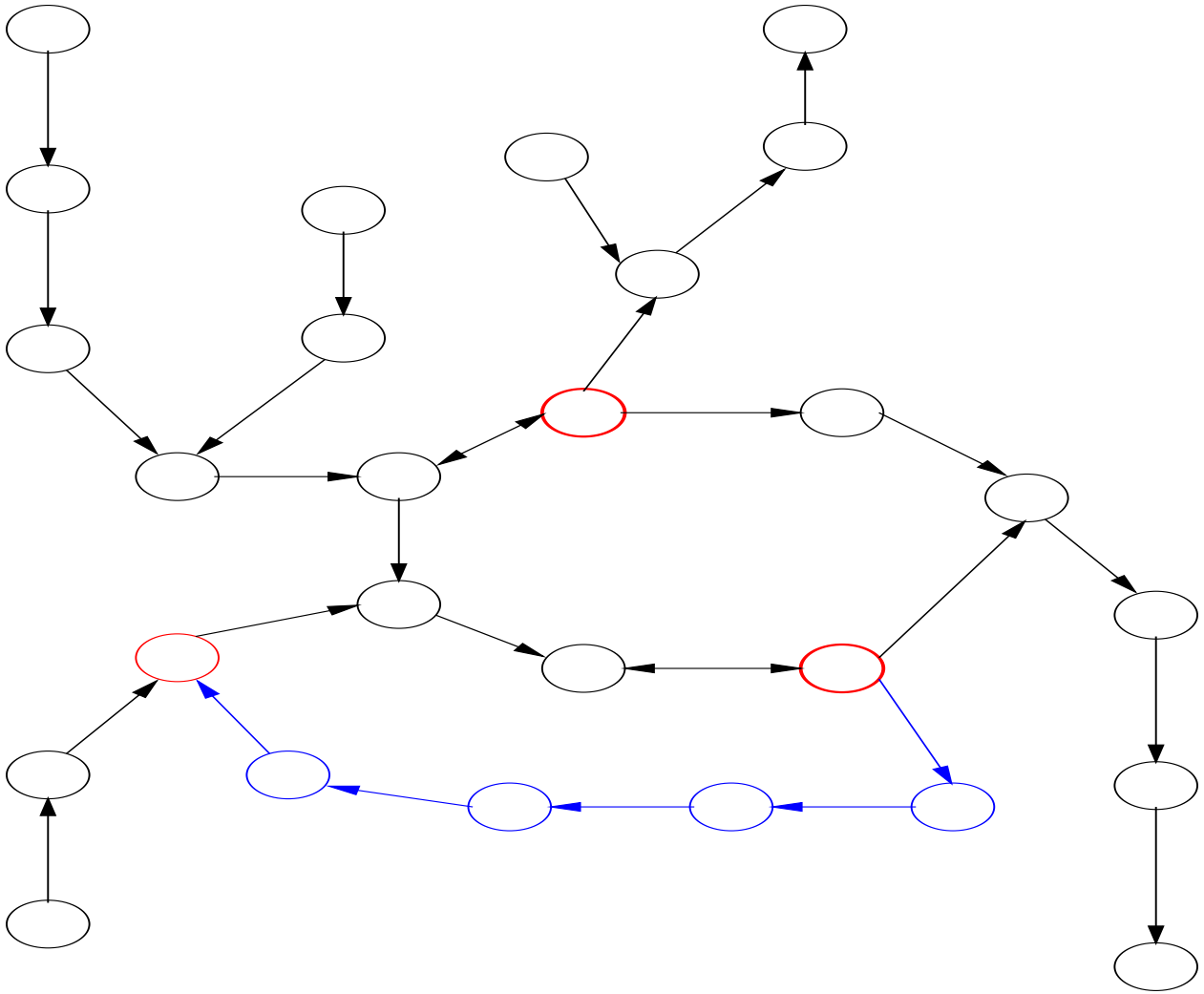
The problem of *Infinite Regress*



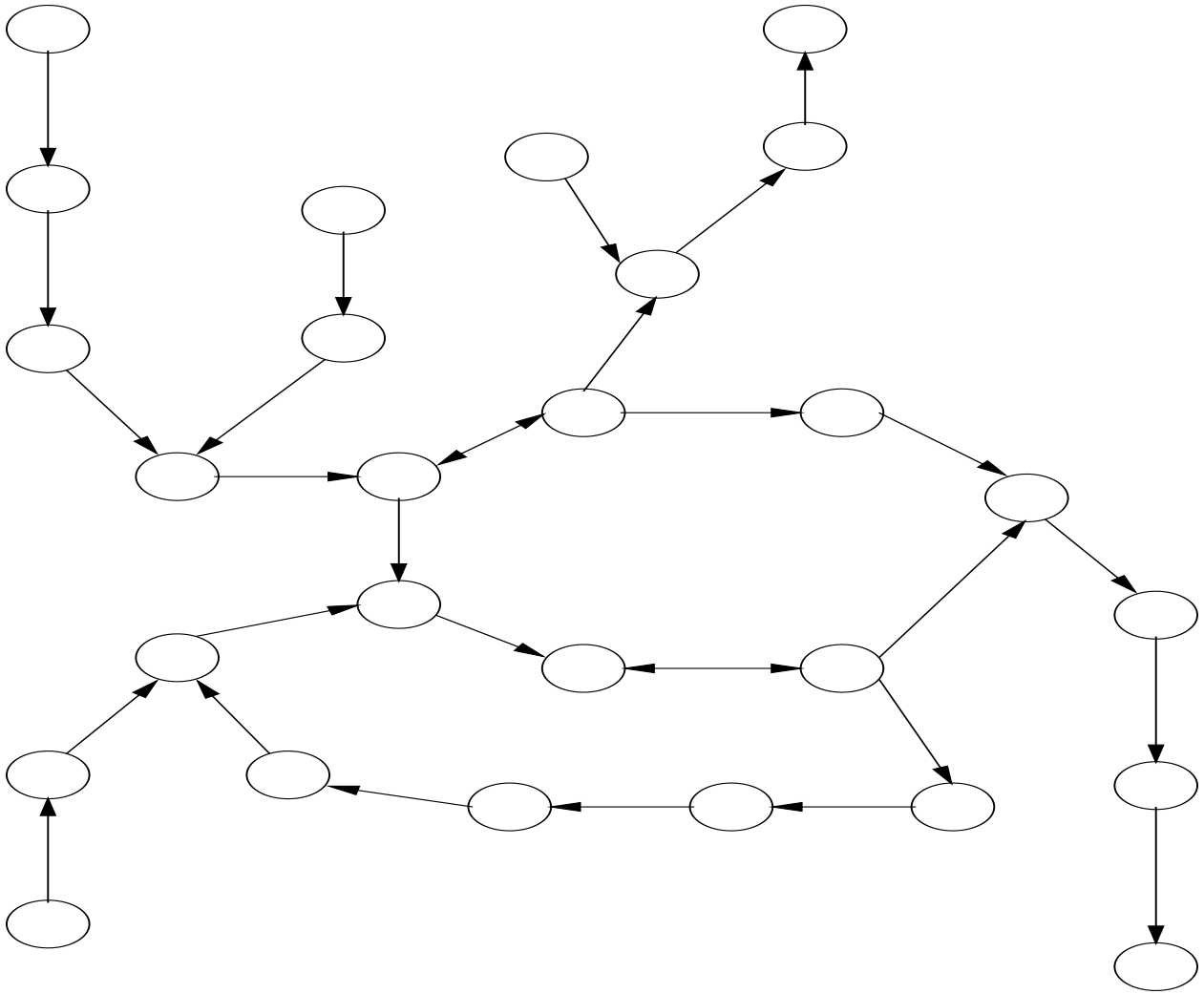
The problem of *Infinite Regress*



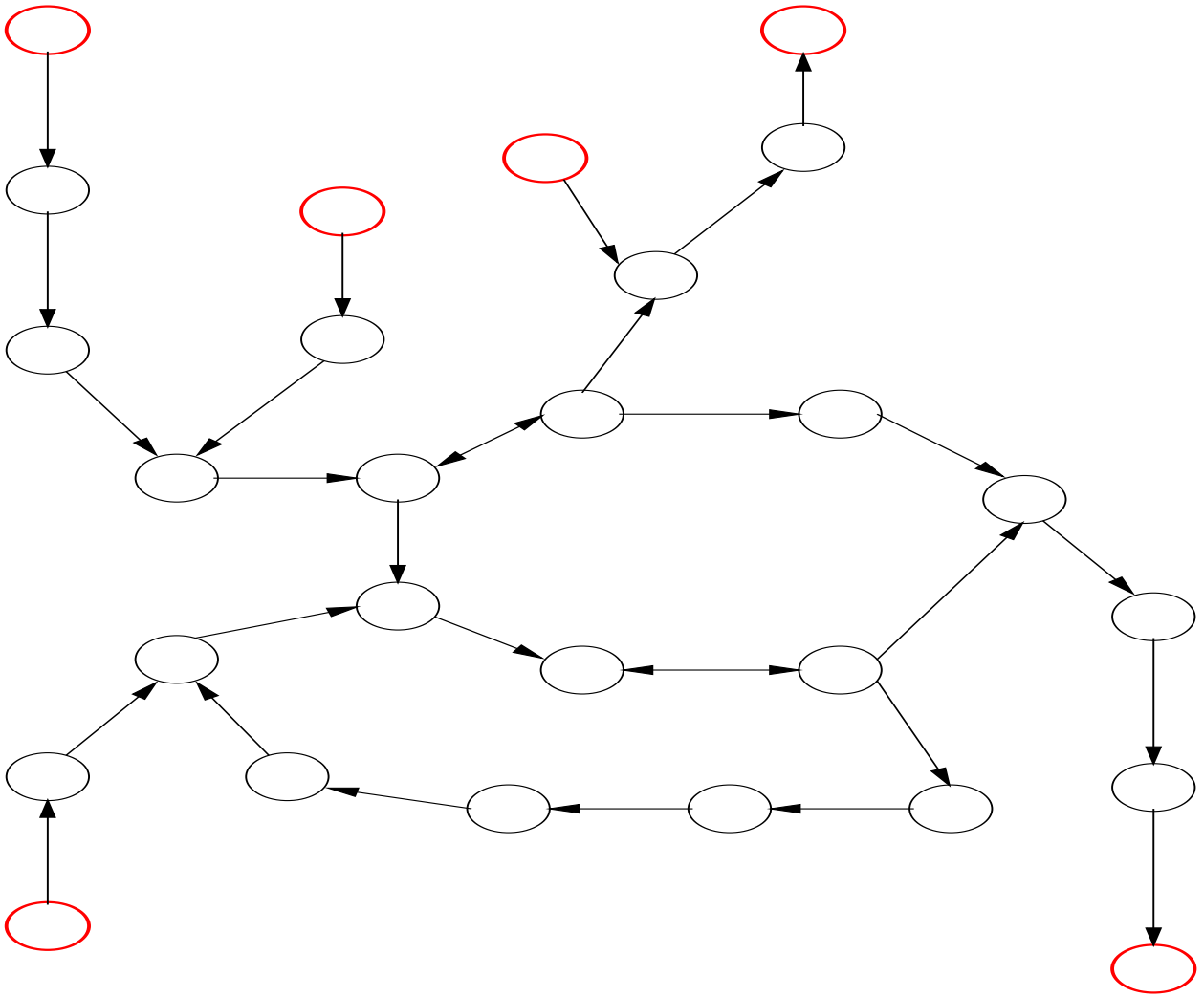
The problem of *Infinite Regress*



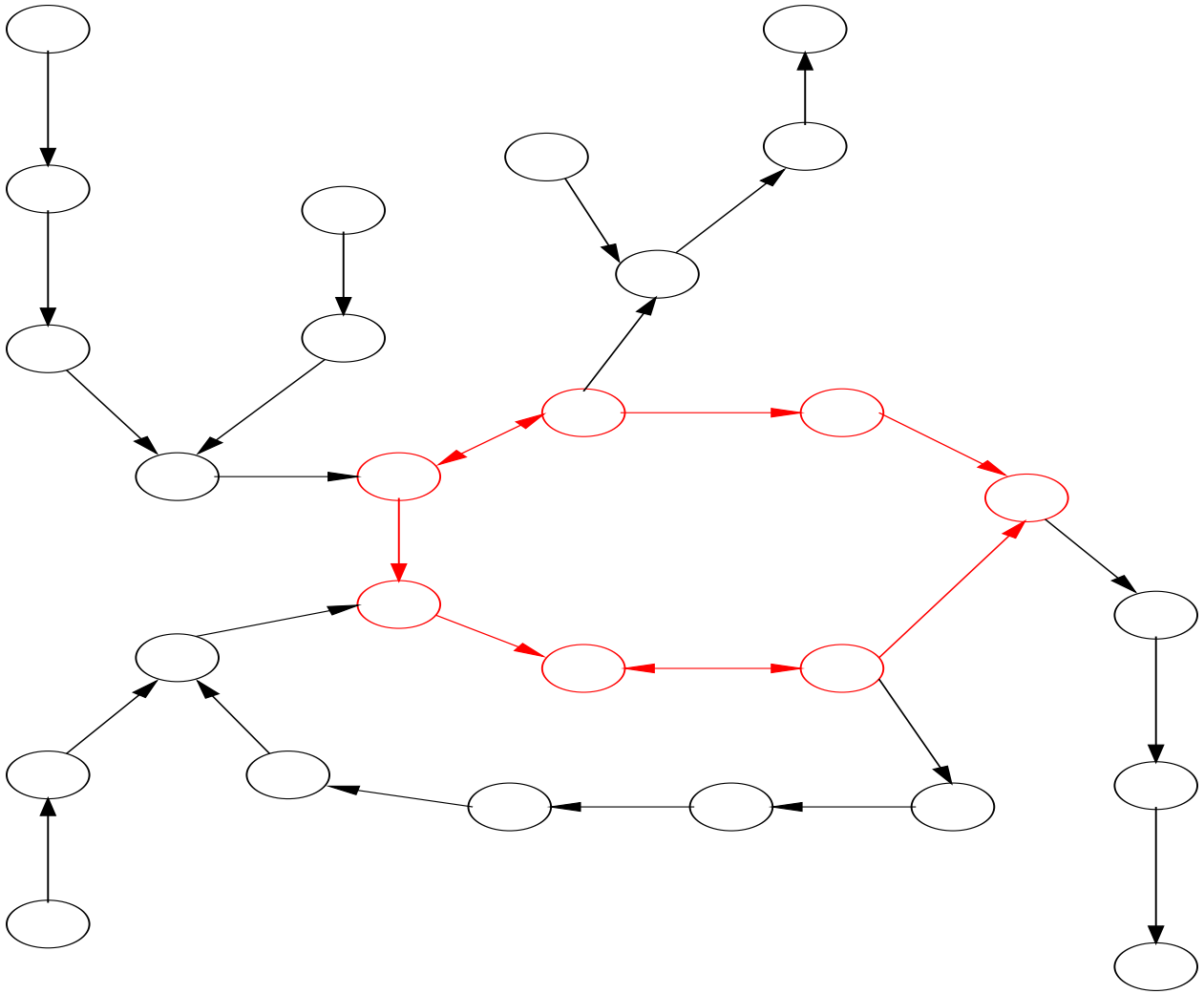
The problem of *Infinite Regress*



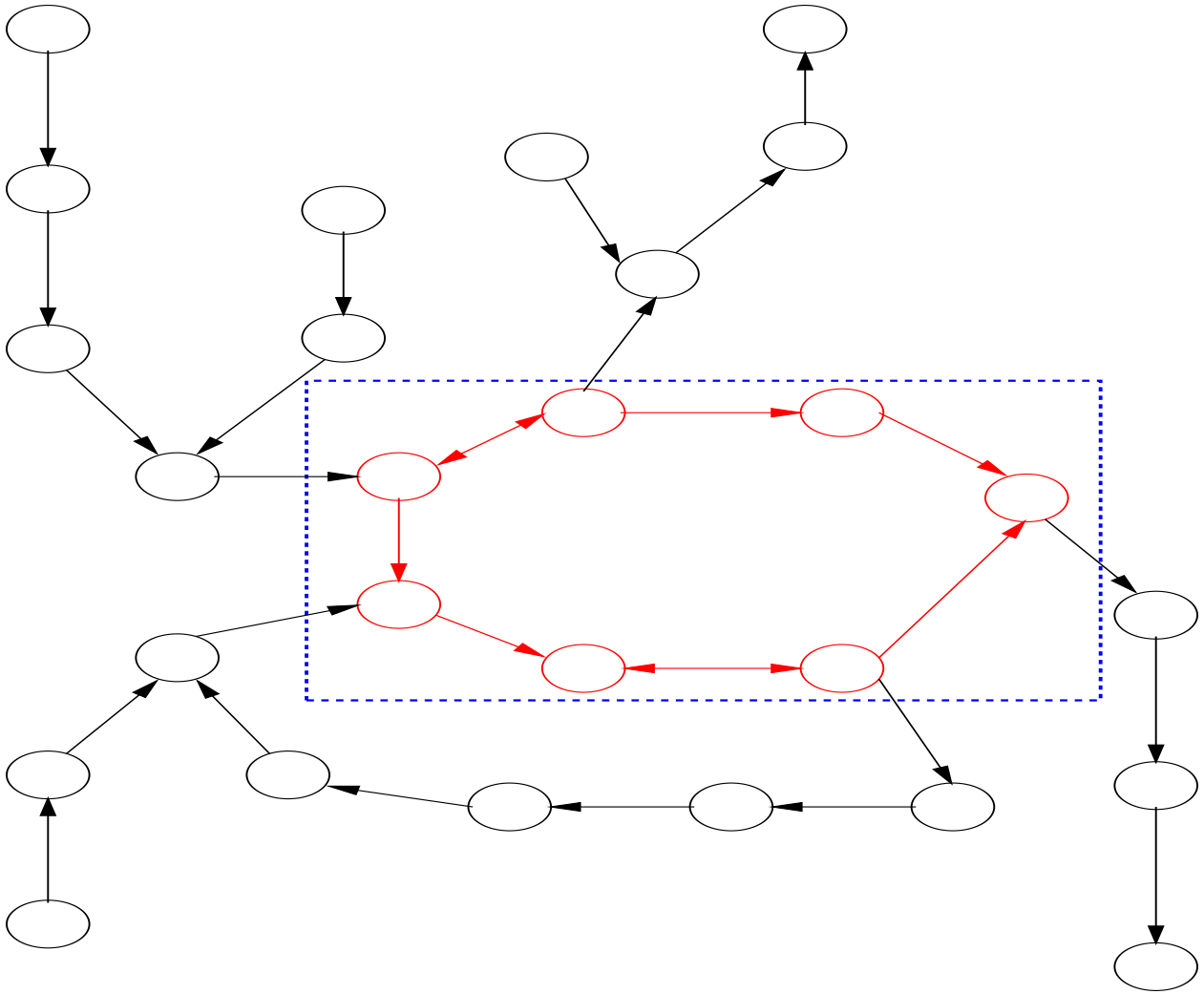
The problem of *Infinite Regress*



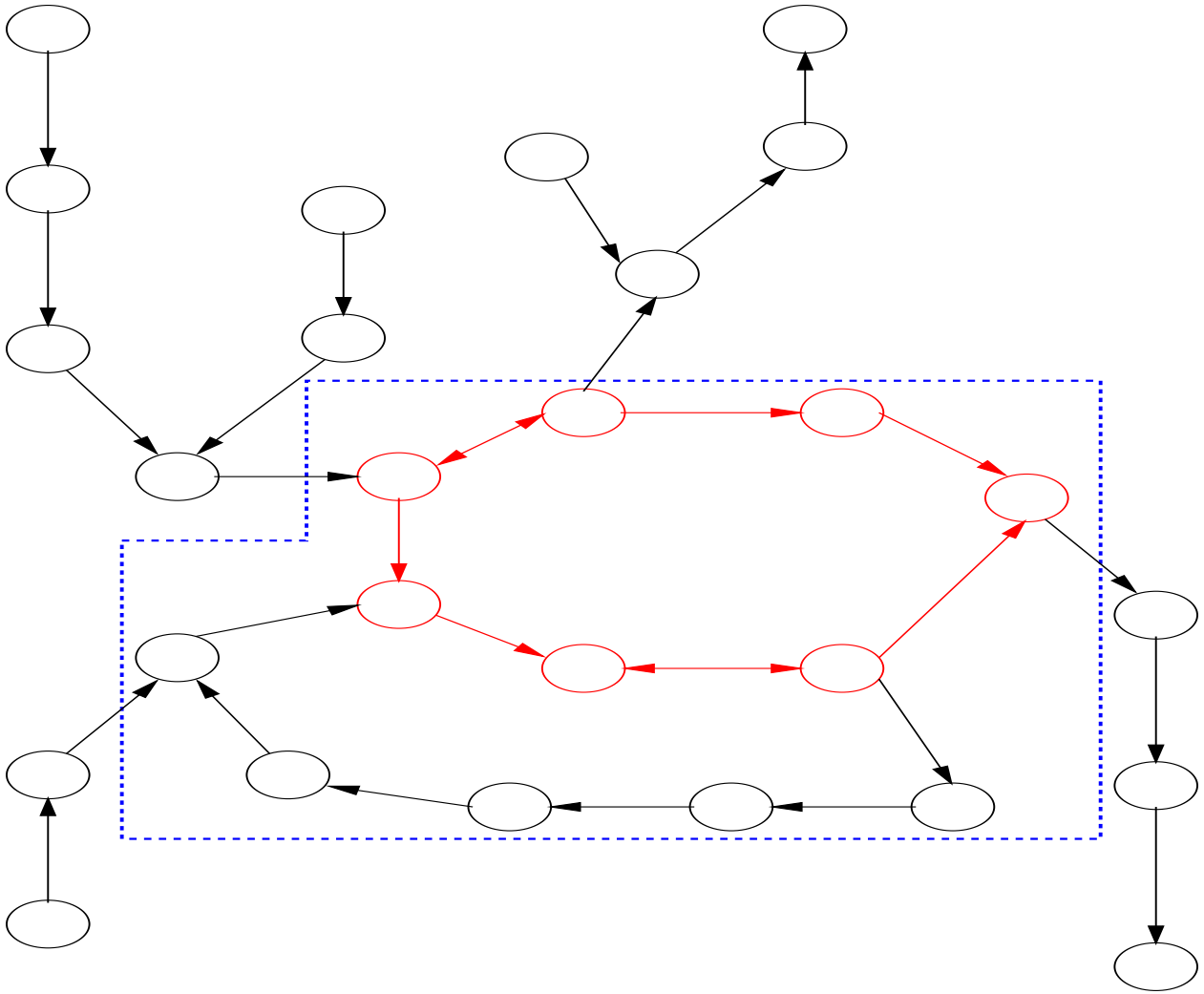
The problem of *Infinite Regress*



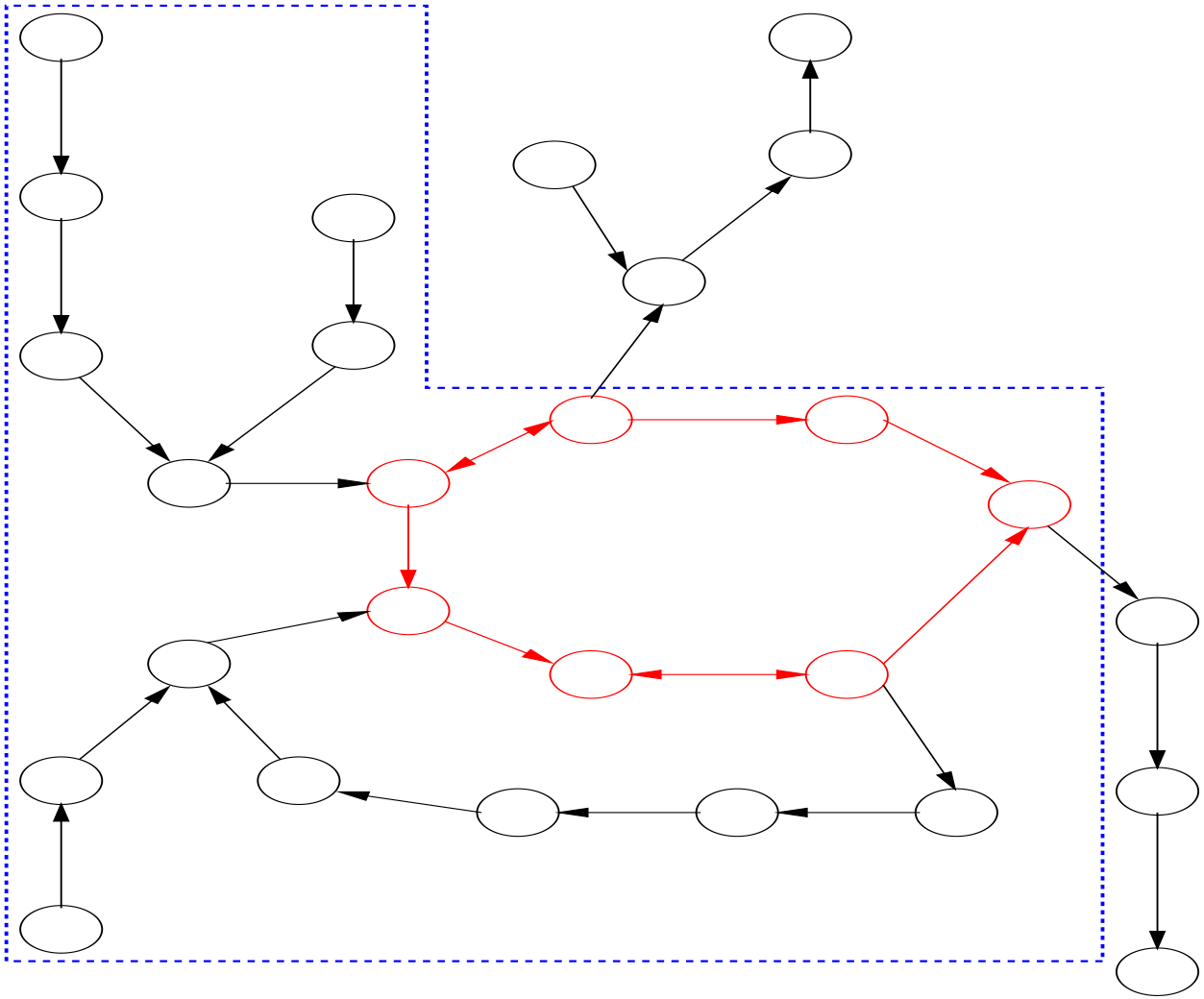
The problem of *Infinite Regress*



The problem of *Infinite Regress*



The problem of *Infinite Regress*



The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.
- A cellular model captures how switching alters the reproductive characteristics of a cell.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.
- A cellular model captures how switching alters the reproductive characteristics of a cell.
- Thus population behaviour depends on cellular behaviour, which is determined by molecular behaviour.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.
- A cellular model captures how switching alters the reproductive characteristics of a cell.
- Thus population behaviour depends on cellular behaviour, which is determined by molecular behaviour.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.
- A cellular model captures how switching alters the reproductive characteristics of a cell.
- Thus population behaviour depends on cellular behaviour, which is determined by molecular behaviour.

The problem of multi-level modelling...

- Some characteristics of systems need to be studied at multiple levels to be fully understood — e.g. *lac* operon in *E. coli*
- A sub-cellular or molecular model only exhibits one type of behaviour.
- A population model is needed to explain the mix of behaviours.
- A cellular model captures how switching alters the reproductive characteristics of a cell.
- Thus population behaviour depends on cellular behaviour, which is determined by molecular behaviour.

A proper account of experimental observations requires a model which captures behaviour at all three levels.

Formal Models

The complexity of biological systems is not fundamentally different from complexity in other forms.

Formal Models

The complexity of biological systems is not fundamentally different from complexity in other forms.

Formal Models

The complexity of computer-based biological models is not fundamentally different from complexity in other computational models.

Formal Models

The complexity of computer-based biological models is not fundamentally different from complexity in other computational models.

Thus many of the techniques we have developed for modelling complex software systems can be beneficially applied to the modelling aspects of systems biology.

Formal Models

The complexity of computer-based biological models is not fundamentally different from complexity in other computational models.

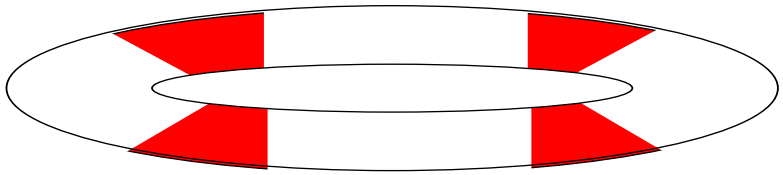
Thus many of the techniques we have developed for modelling complex software systems can be beneficially applied to the modelling aspects of systems biology.

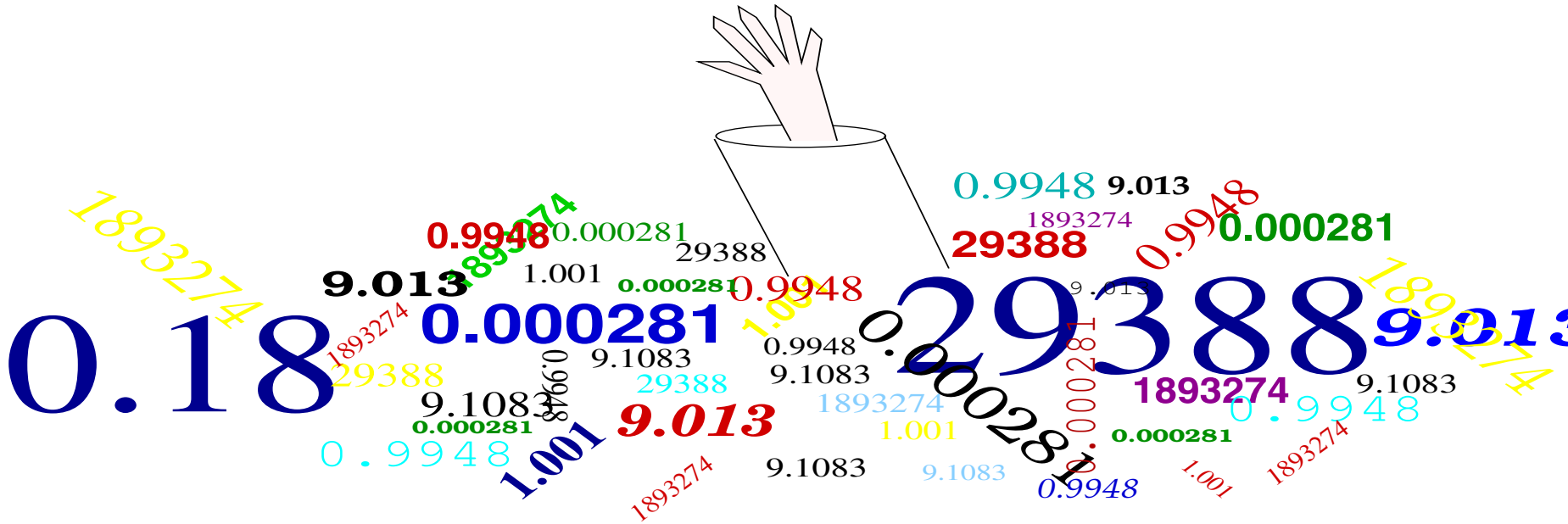
In particular:

- Abstraction
- Modularity and
- Reasoning

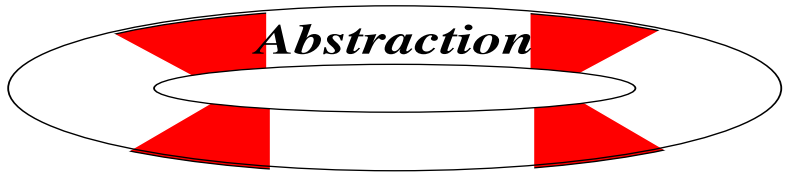
have a key role to play.

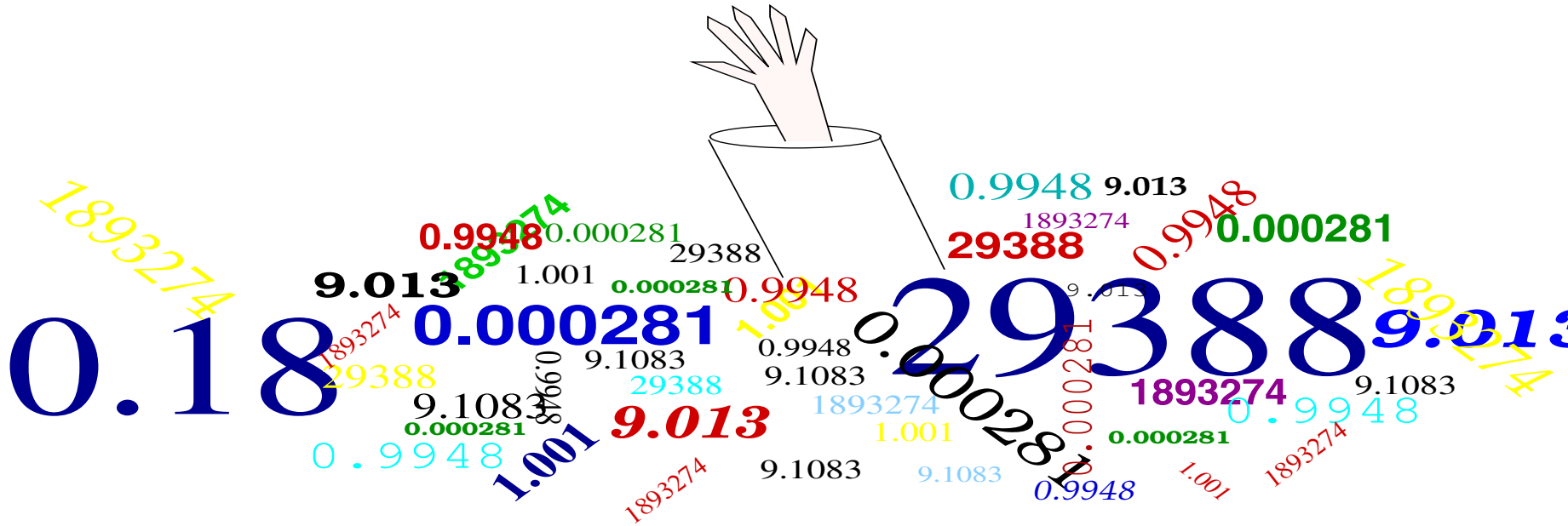
The Role of Computational Thinking



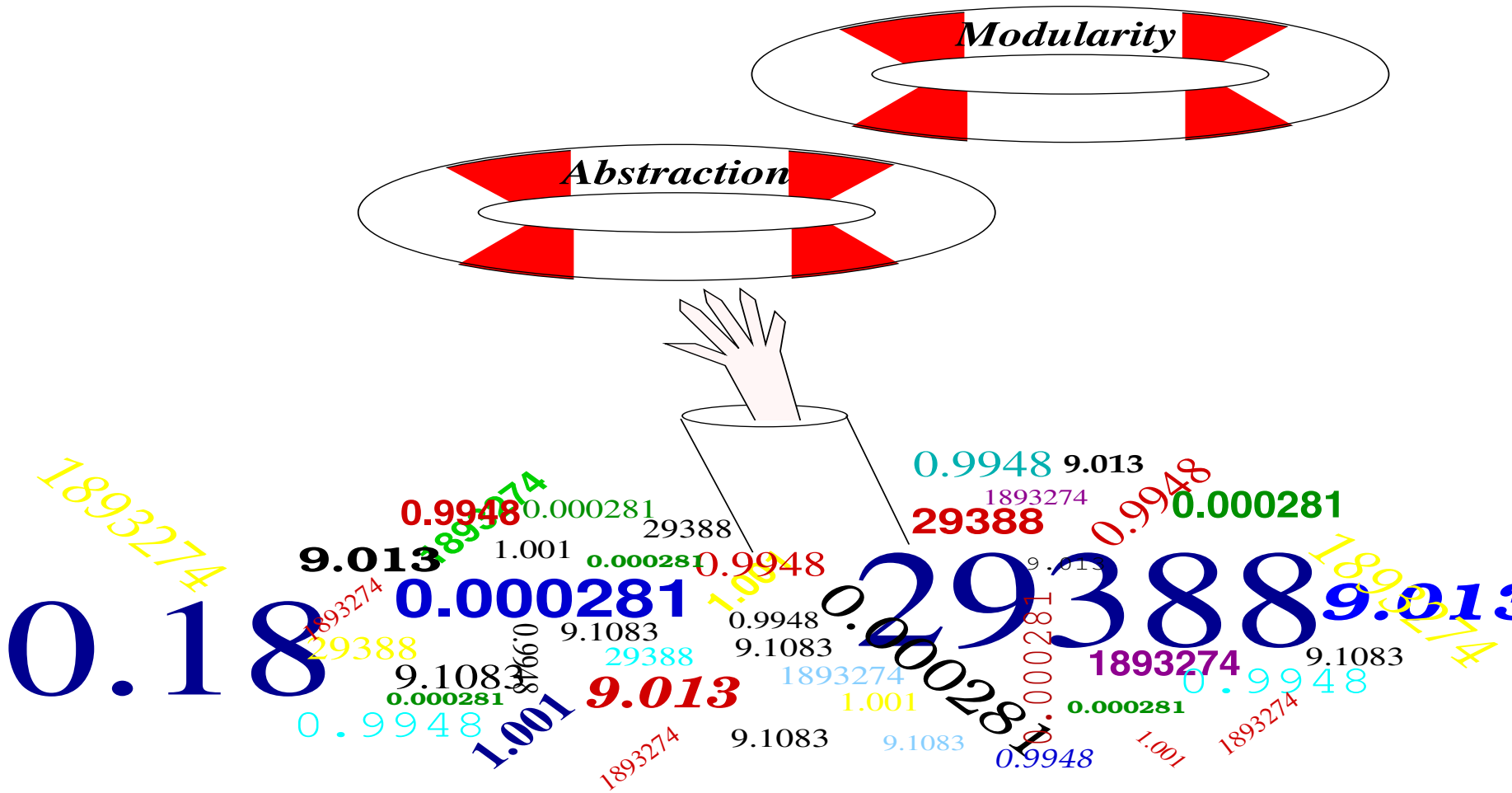


The Role of Computational Thinking

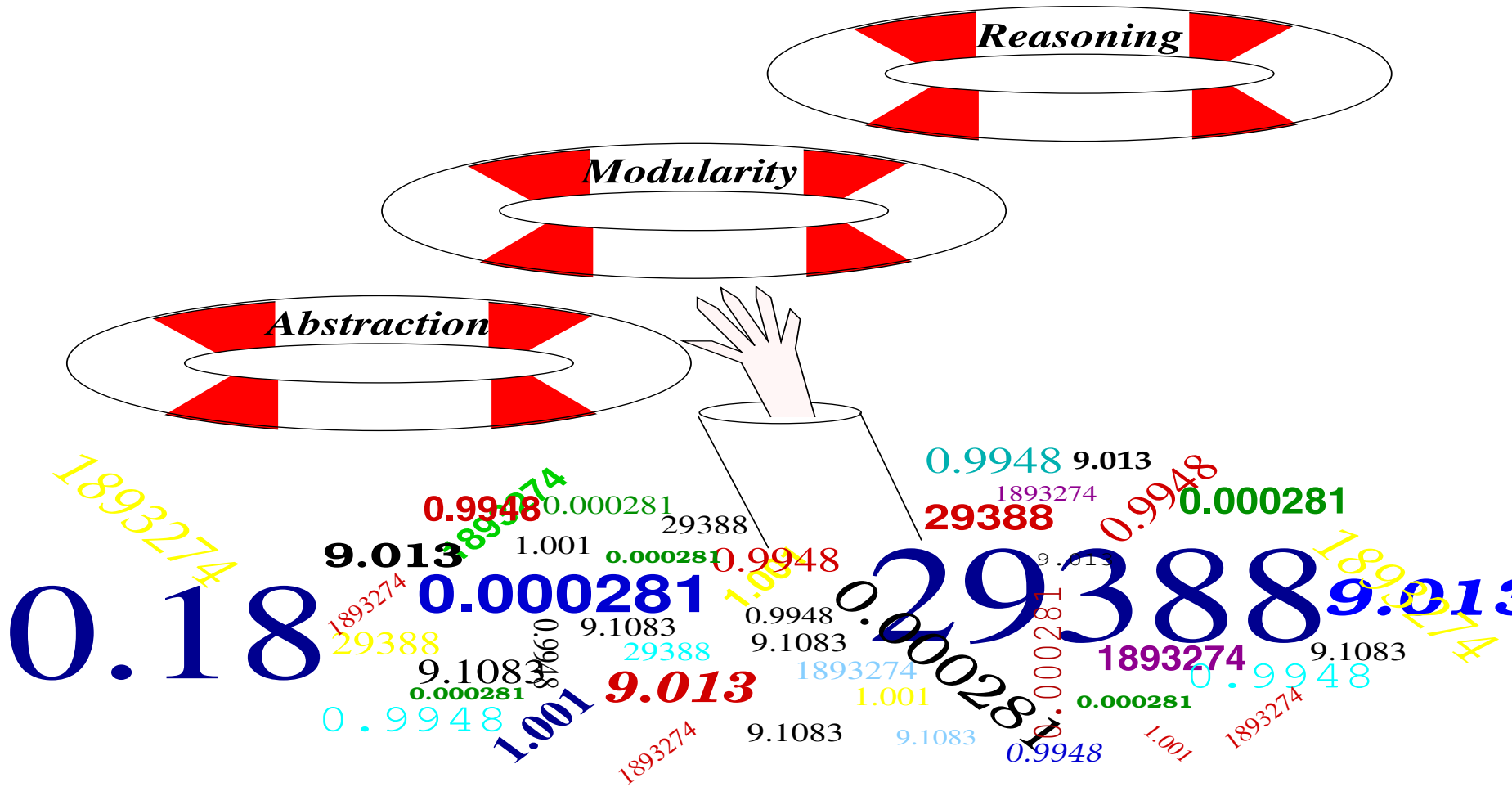




The Role of Computational Thinking



The Role of Computational Thinking



Outline

Formal Models for Systems Biology

When systems biology was emerging in the 1950s and 1960s the role of computers, and computational thinking, was confined to system analysis (largely simulation).

Formal Models for Systems Biology

When systems biology was emerging in the 1950s and 1960s the role of computers, and computational thinking, was confined to system analysis (largely simulation).

In the intervening period substantial developments have been made in theoretical computer science with respect to formal system description techniques.

Formal Models for Systems Biology

When systems biology was emerging in the 1950s and 1960s the role of computers, and computational thinking, was confined to system analysis (largely simulation).

In the intervening period substantial developments have been made in theoretical computer science with respect to formal system description techniques.

With the current explosion of interest in systems biology the application of many of theses techniques to biological systems has been explored.

Formal Models for Systems Biology

When systems biology was emerging in the 1950s and 1960s the role of computers, and computational thinking, was confined to system analysis (largely simulation).

In the intervening period substantial developments have been made in theoretical computer science with respect to formal system description techniques.

With the current explosion of interest in systems biology the application of many of theses techniques to biological systems has been explored.

I will focus on the use of process algebras for signalling pathways within cells.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.
- More complex behaviours are built up from interactions between components; concurrency is assumed.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.
- More complex behaviours are built up from interactions between components; concurrency is assumed.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.
- More complex behaviours are built up from interactions between components; concurrency is assumed.
- Thus process algebraic formulations make interactions/constraints explicit; structure can also be apparent.

Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.
- More complex behaviours are built up from interactions between components; concurrency is assumed.
- Thus process algebraic formulations make interactions/constraints explicit; structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.

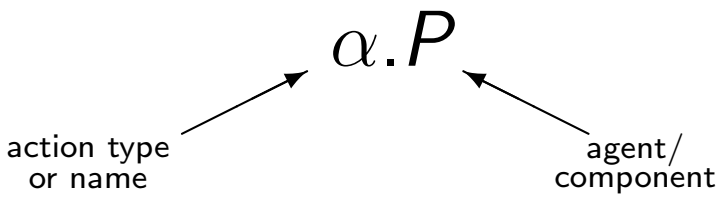
Process Algebras for Systems Biology

Process algebras have several attractive features which can be useful for modelling and understanding biological systems:

- The primitives of the formalism are agents or components which engage in activities.
- More complex behaviours are built up from interactions between components; concurrency is assumed.
- Thus process algebraic formulations make interactions/constraints explicit; structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.
- There are well-established techniques for reasoning about the behaviours and properties of models, supported by software.

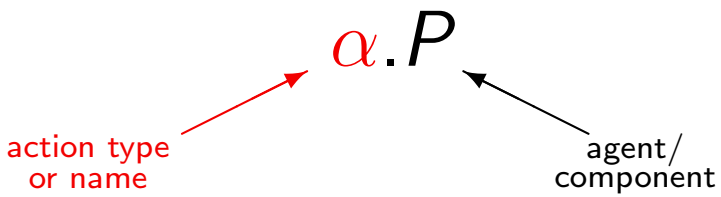
Process Algebra

- Models consist of agents which engage in actions.



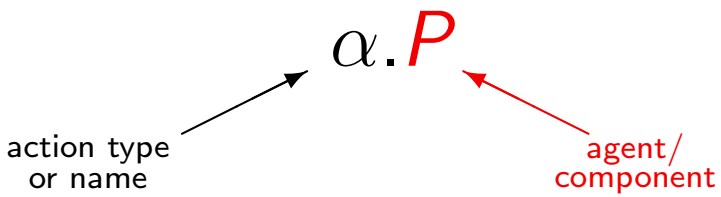
Process Algebra

- Models consist of agents which engage in actions.



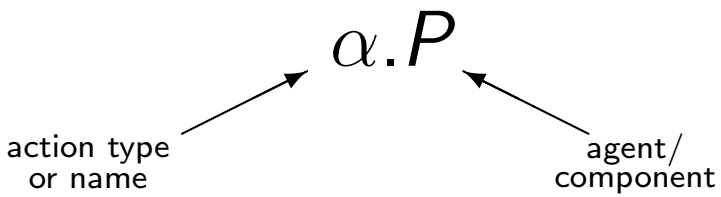
Process Algebra

- Models consist of agents which engage in actions.



Process Algebra

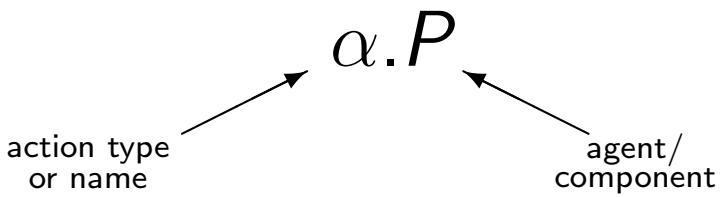
- Models consist of agents which engage in actions.



- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process Algebra

- Models consist of agents which engage in actions.

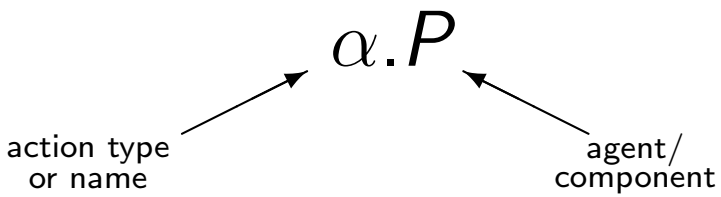


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model

Process Algebra

- Models consist of agents which engage in actions.

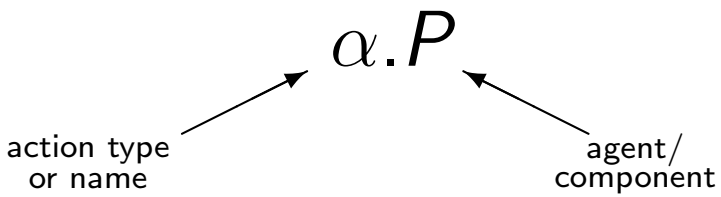


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model $\xrightarrow{\text{SOS rules}}$

Process Algebra

- Models consist of agents which engage in actions.

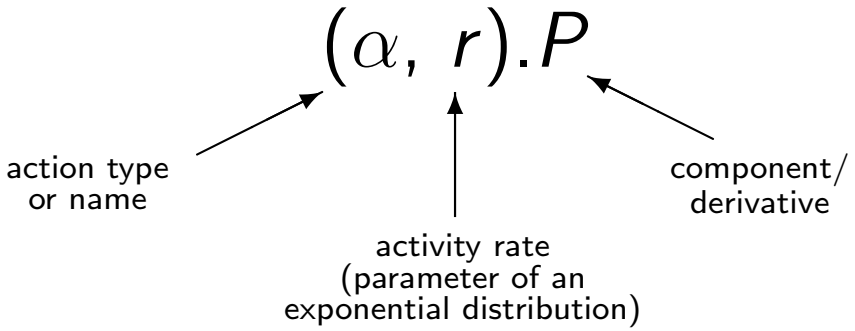


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model $\xrightarrow{\text{SOS rules}}$ Labelled transition system

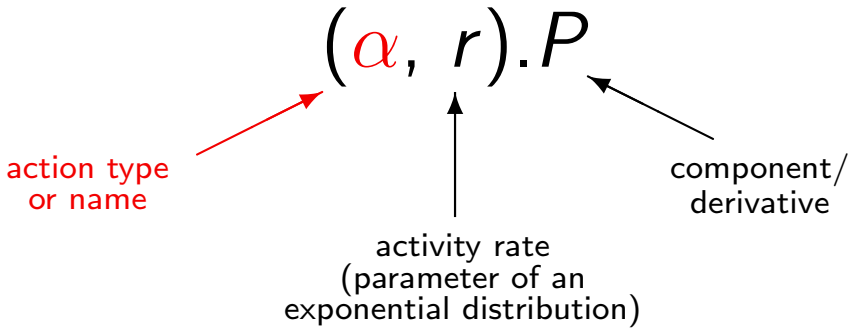
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



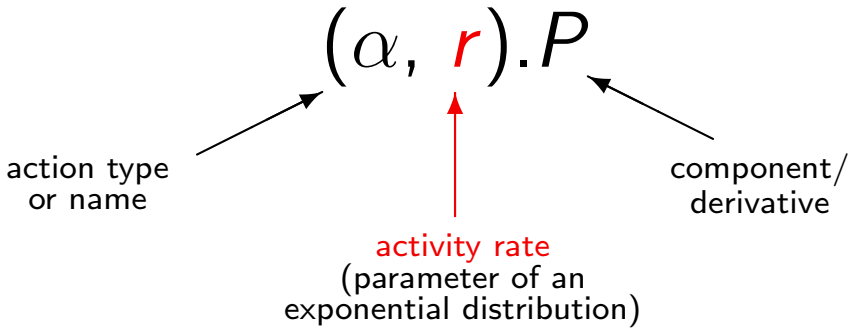
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



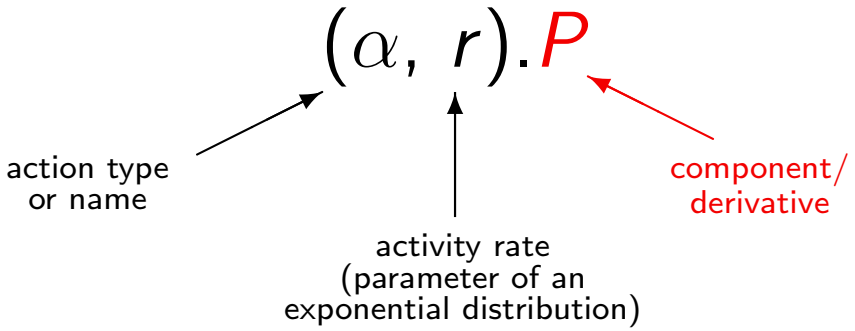
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



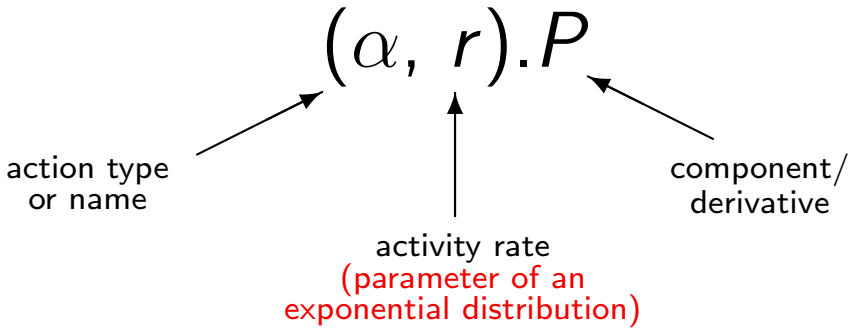
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



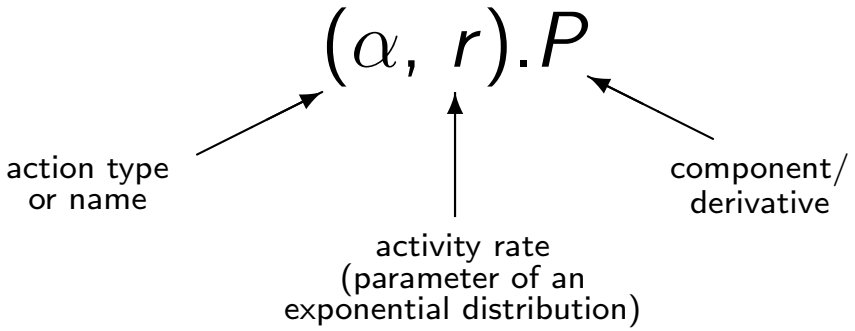
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



Stochastic Process Algebra

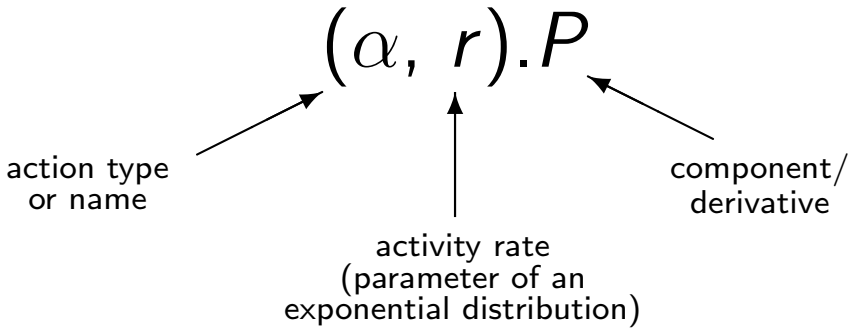
- Models are constructed from components which engage in activities.



The language may be used to generate a Markov Process (CTMC).

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

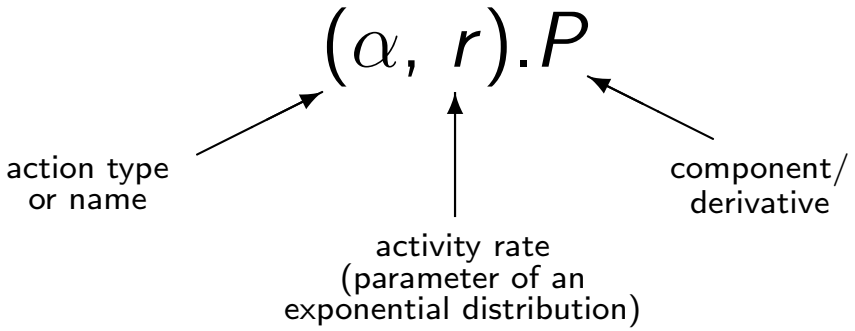


The language may be used to generate a Markov Process (CTMC).

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

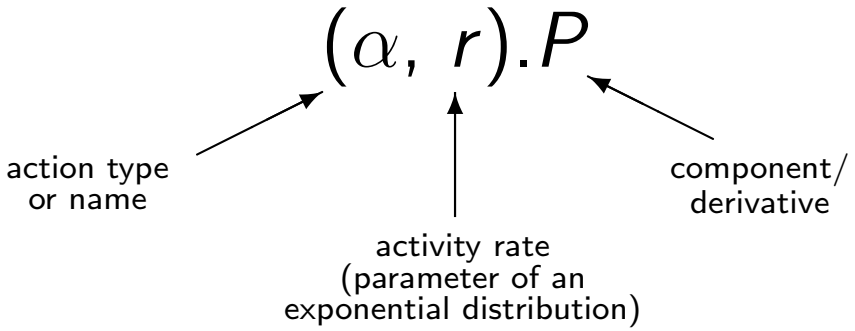


The language may be used to generate a Markov Process (CTMC).

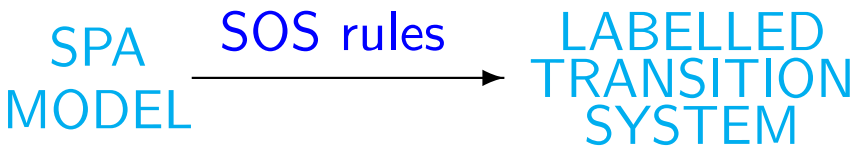
SPA MODEL $\xrightarrow{\text{SOS rules}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

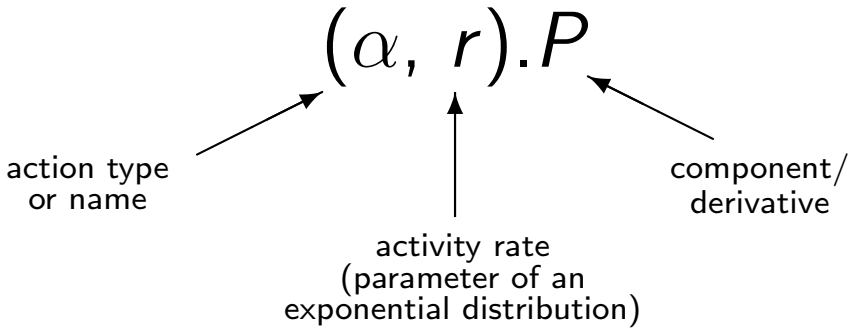


The language may be used to generate a Markov Process (CTMC).



Stochastic Process Algebra

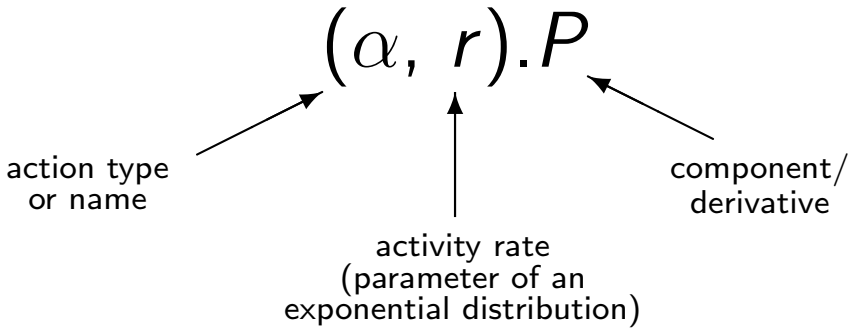
- Models are constructed from components which engage in activities.



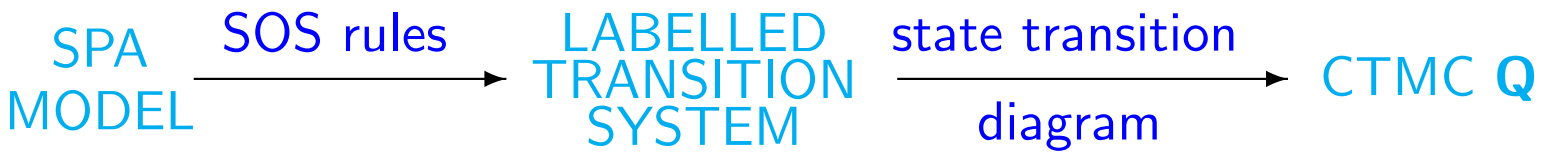
The language may be used to generate a Markov Process (CTMC).

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

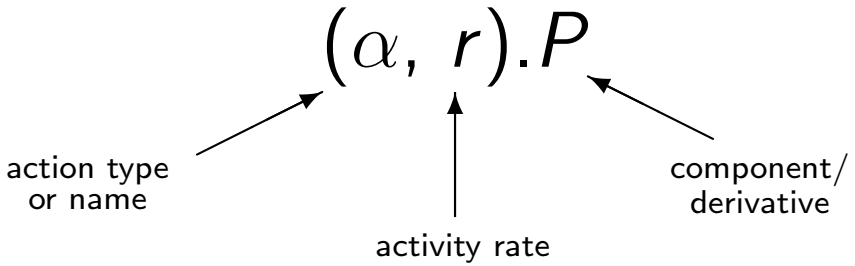


The language may be used to generate a Markov Process (CTMC).



Stochastic Process Algebra

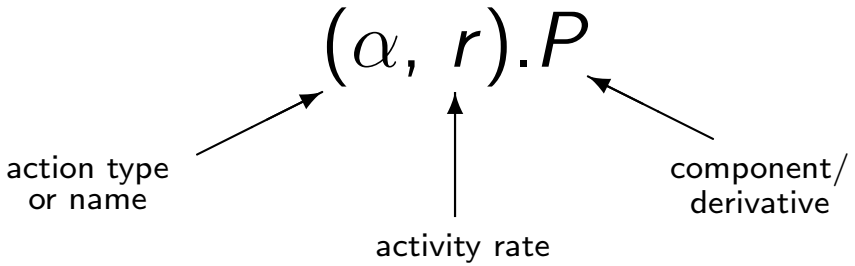
- Models are constructed from components which engage in activities.



The language may be used to generate a system of ordinary differential equations (ODEs).

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

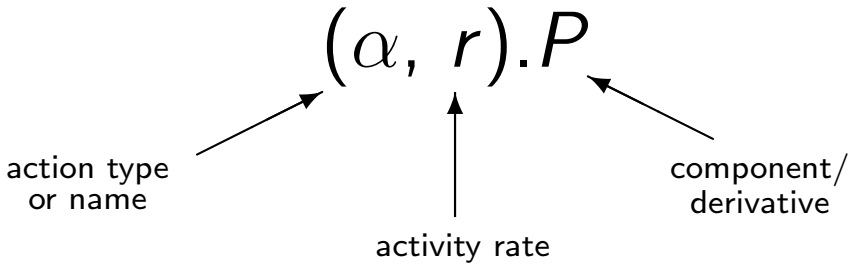


The language may be used to generate a system of ordinary differential equations (ODEs).

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

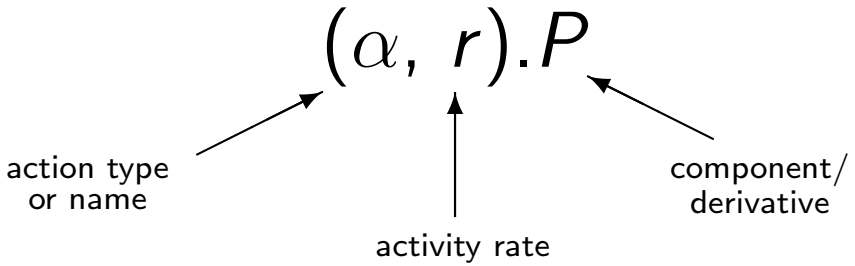


The language may be used to generate a system of ordinary differential equations (ODEs).

SPA MODEL $\xrightarrow[\text{analysis}]{\text{syntactic}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

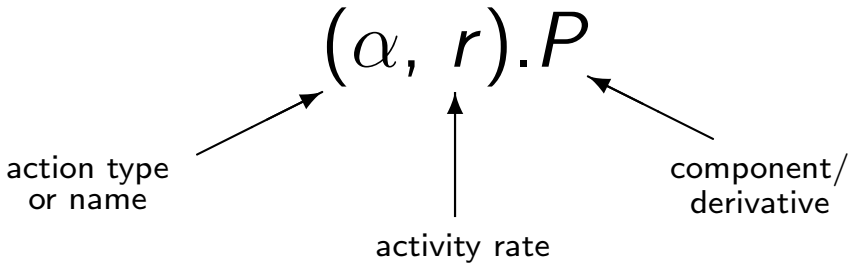


The language may be used to generate a system of ordinary differential equations (ODEs).

SPA MODEL —syntactic analysis→ ACTIVITY MATRIX

Stochastic Process Algebra

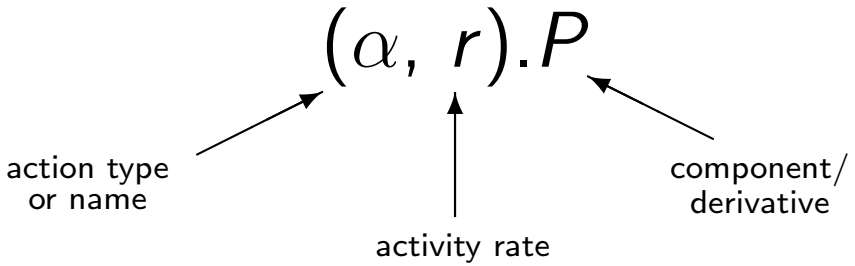
- Models are constructed from components which engage in activities.



The language may be used to generate a system of ordinary differential equations (ODEs).

Stochastic Process Algebra

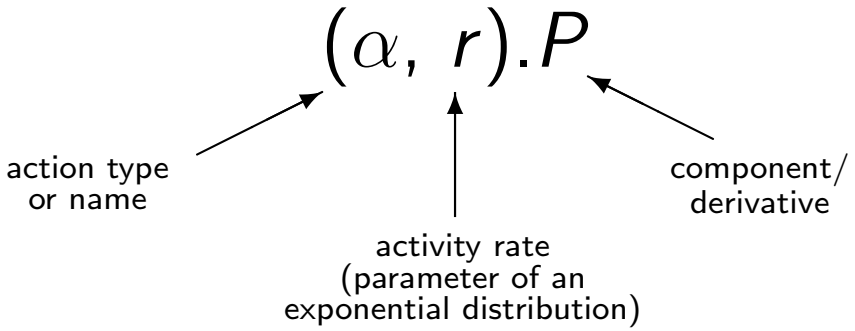
- Models are constructed from components which engage in activities.



The language may be used to generate a system of ordinary differential equations (ODEs).

Stochastic Process Algebra

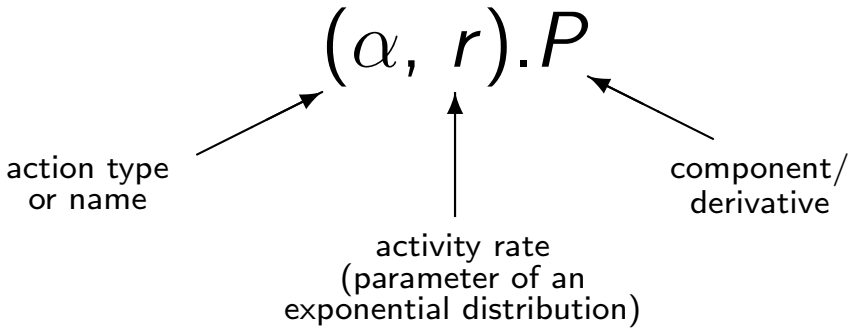
- Models are constructed from components which engage in activities.



The language also may be used to generate a stochastic simulation.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

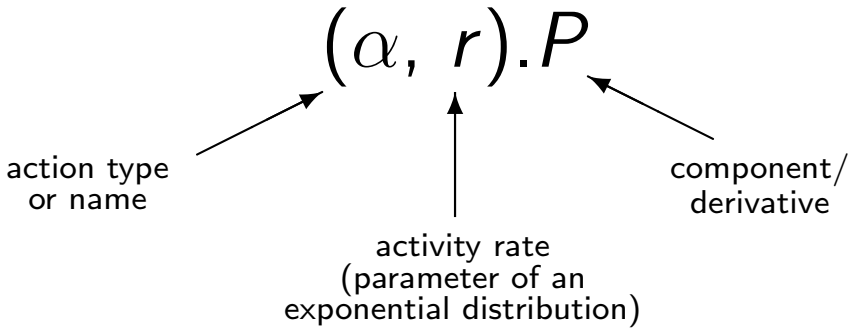


The language also may be used to generate a stochastic simulation.

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

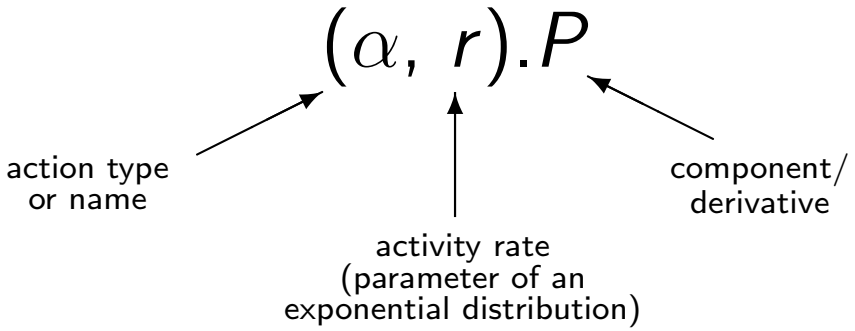


The language also may be used to generate a stochastic simulation.

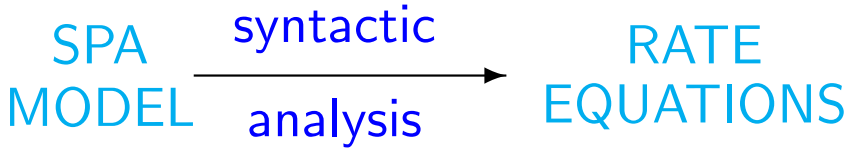
SPA MODEL —syntactic analysis→

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

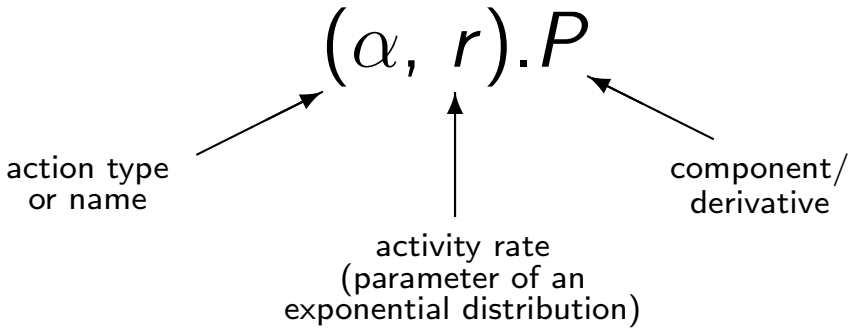


The language also may be used to generate a stochastic simulation.



Stochastic Process Algebra

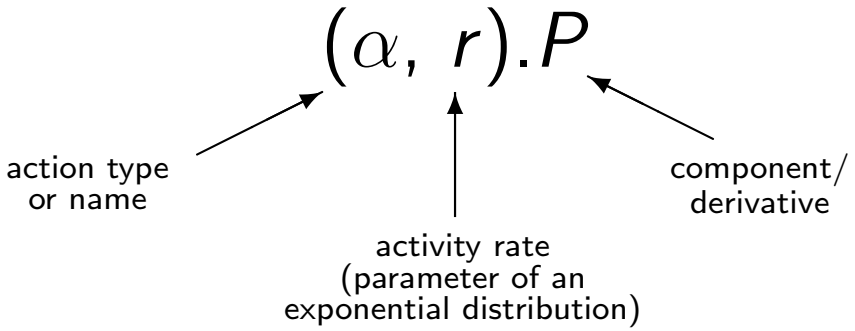
- Models are constructed from components which engage in activities.



The language also may be used to generate a stochastic simulation.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



The language also may be used to generate a stochastic simulation.

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Alternative Mappings

In the PEPA modelling we have been doing we have experimented with more abstract mappings between process algebra constructs and elements of signal transduction pathways.

Alternative Mappings

In the PEPA modelling we have been doing we have experimented with more abstract mappings between process algebra constructs and elements of signal transduction pathways.

In particular we consider alternatives to the molecule as the basic building block.

Alternative Mappings

In the PEPA modelling we have been doing we have experimented with more abstract mappings between process algebra constructs and elements of signal transduction pathways.

In particular we consider alternatives to the molecule as the basic building block.

In our first mapping we focus on species (c.f. a type rather than an instance, or a class rather than an object).

Alternative Mappings

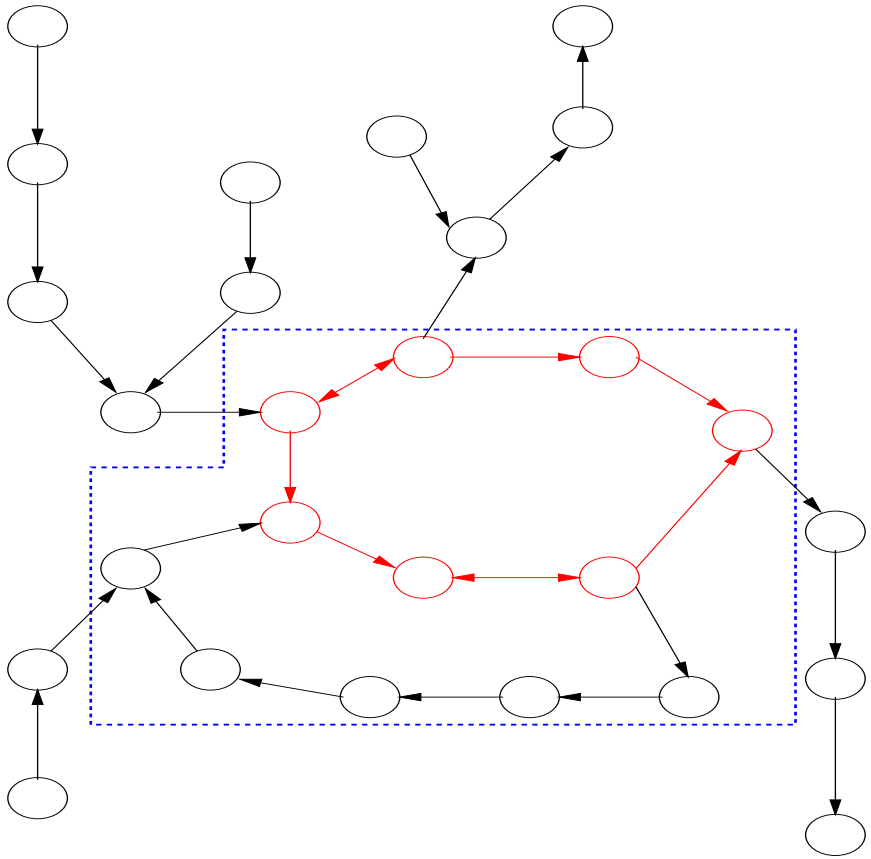
In the PEPA modelling we have been doing we have experimented with more abstract mappings between process algebra constructs and elements of signal transduction pathways.

In particular we consider alternatives to the molecule as the basic building block.

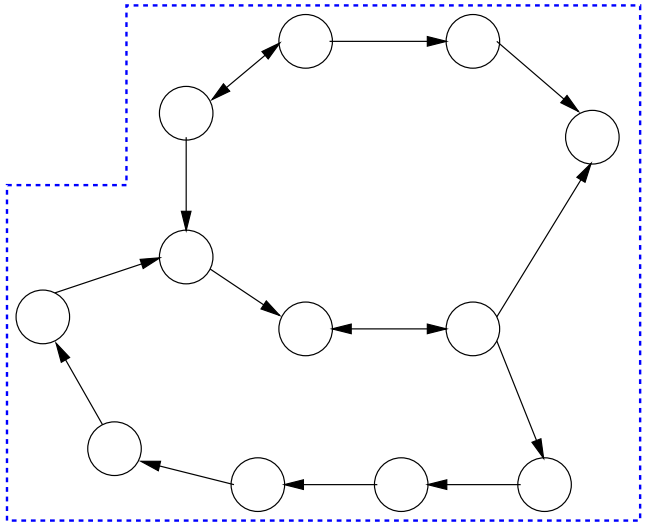
In our first mapping we focus on species (c.f. a type rather than an instance, or a class rather than an object).

In our second we focus on sub-pathways.

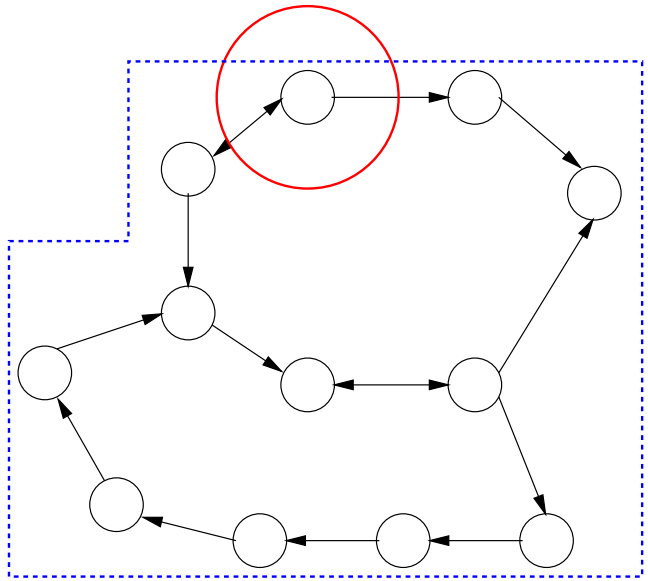
Alternative Mappings illustration



Alternative Mappings illustration

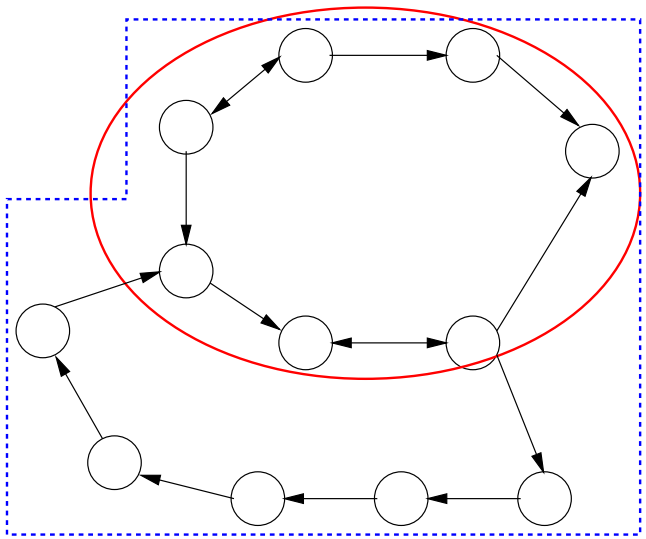


Alternative Mappings illustration



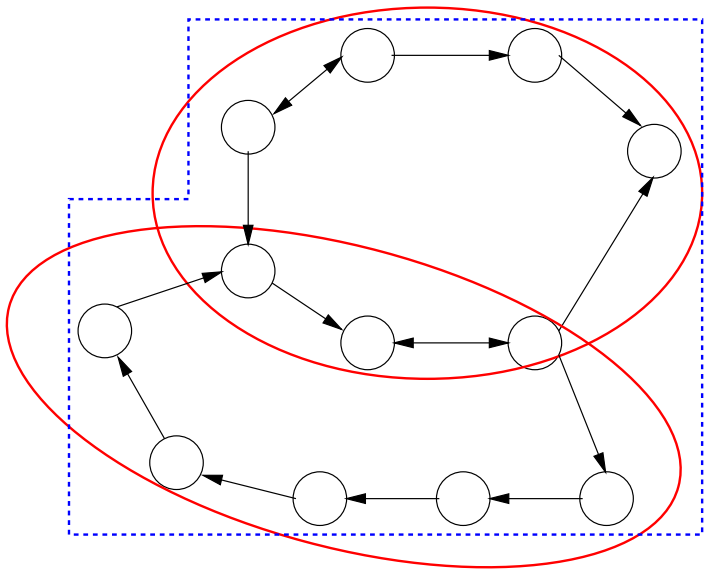
Reagent mapping: Each species is a distinct component in the model with local states to capture differing levels of concentration

Alternative Mappings illustration



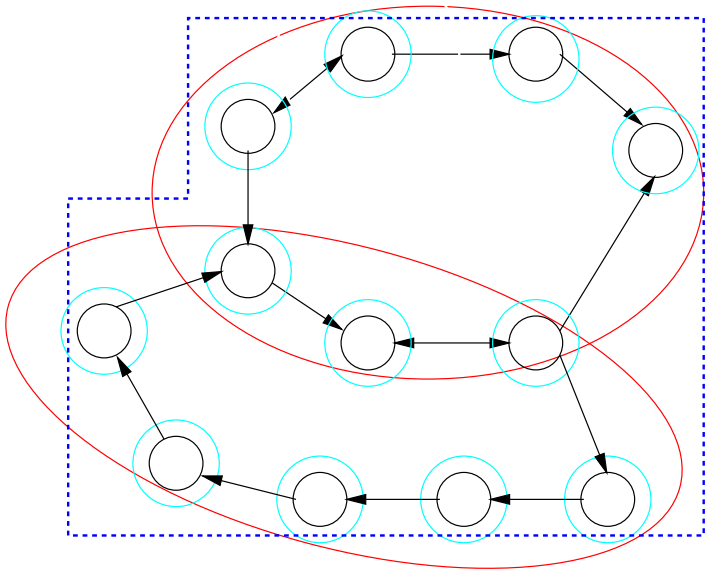
Pathway mapping: Each sub-pathway is a distinct component in the model with local states to capture progress through the pathway

Alternative Mappings illustration



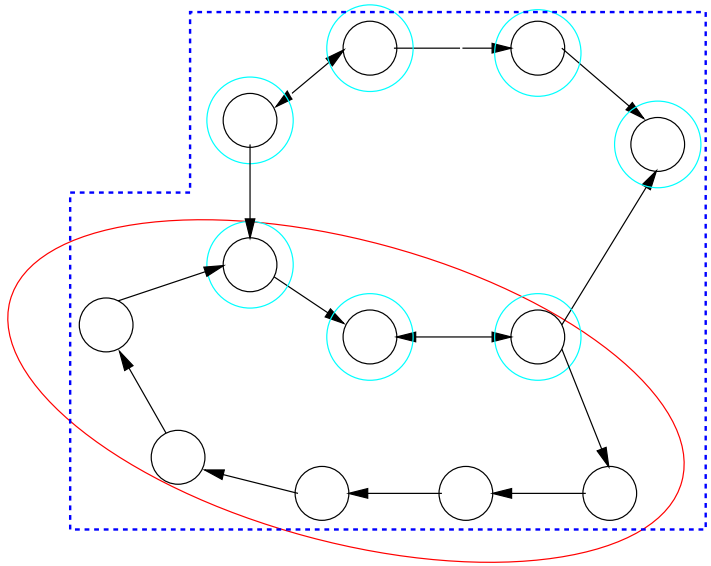
Pathway mapping: Each sub-pathway is a distinct component in the model with local states to capture progress through the pathway

Alternative Mappings illustration



Reasoning based on bisimulation equivalence is able to prove that the two representation are equivalent.

Alternative Mappings illustration



Different parts of the system may use different mappings, reflecting perhaps the level of knowledge (data) available, or the primary interests of the modeller.

Abstraction

- Process algebras offer abstraction in both their style of modelling, and as a formal operation which can be applied to models after construction (e.g. *hiding* or *restriction*).

Abstraction

- Process algebras offer abstraction in both their style of modelling, and as a formal operation which can be applied to models after construction (e.g. *hiding* or *restriction*).
- Our aim when modelling a system is to capture sufficient information to be able to carry out useful (quantitative) analysis — not necessary to create the most faithful representation of the system possible.

Abstraction

- Process algebras offer abstraction in both their style of modelling, and as a formal operation which can be applied to models after construction (e.g. *hiding* or *restriction*).
- Our aim when modelling a system is to capture sufficient information to be able to carry out useful (quantitative) analysis — not necessary to create the most faithful representation of the system possible.
- Suitable equivalence relations can confirm that our abstraction is valid.

Modularity

- Compositionality is an inherent feature of process algebras giving all such models modularity.

Modularity

- Compositionality is an inherent feature of process algebras giving all such models modularity.
- As well as the clear advantages that this has for model construction (c.f. software engineering), it also offers potential benefits for multi-level modelling.

Modularity

- Compositionality is an inherent feature of process algebras giving all such models modularity.
- As well as the clear advantages that this has for model construction (c.f. software engineering), it also offers potential benefits for multi-level modelling.
- Moreover, in the Markovian setting, work has already been done to identify forms of interaction in a process algebra which are amenable to decomposed quantitative analysis.

Reasoning

- Process algebras are equipped with equivalence relations, and partial relations.

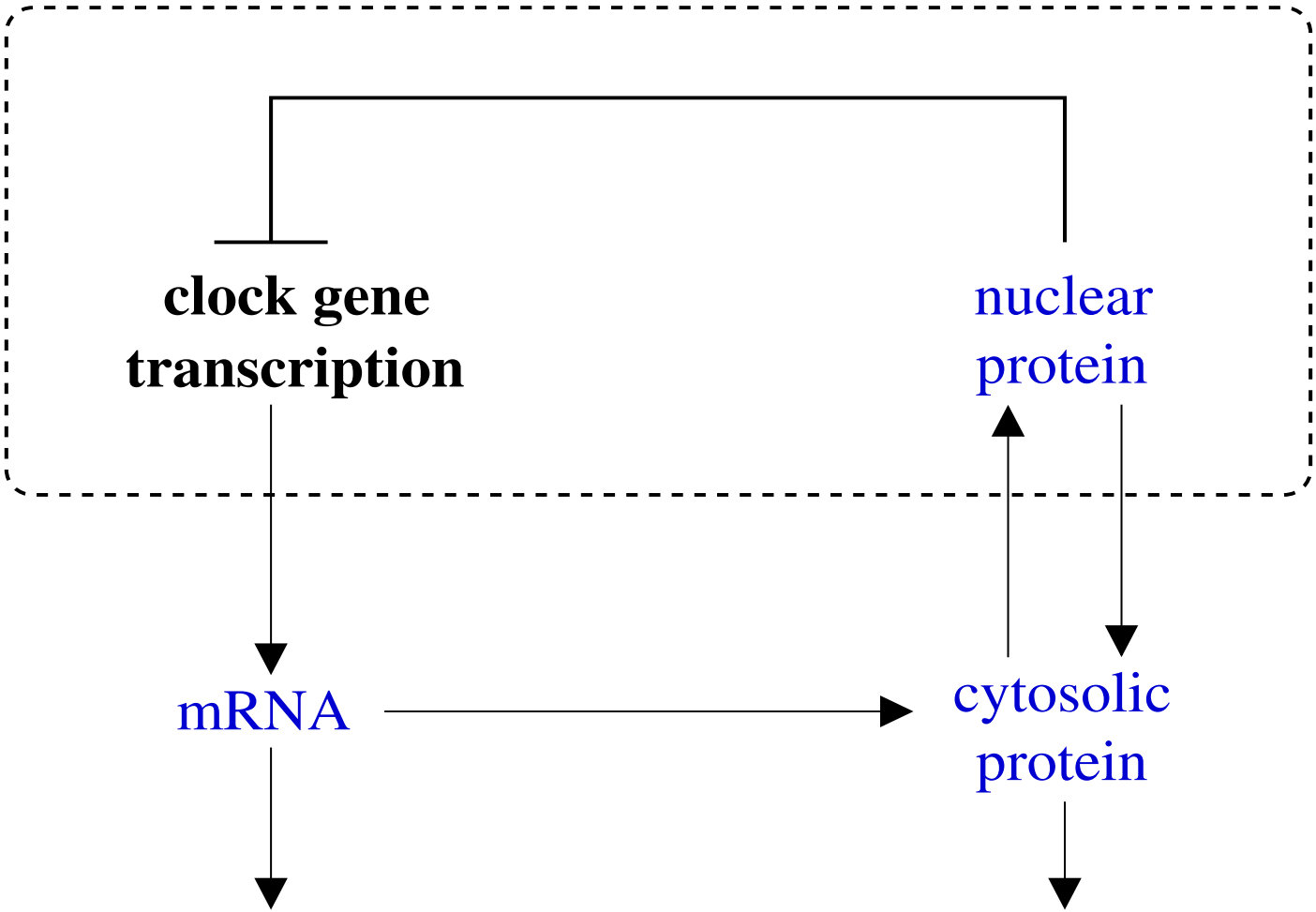
Reasoning

- Process algebras are equipped with equivalence relations, and partial relations.
- These allow reasoning about the relationships between models: either alternative representations (as we have seen) or models which result from simplification or elaboration of an original model.

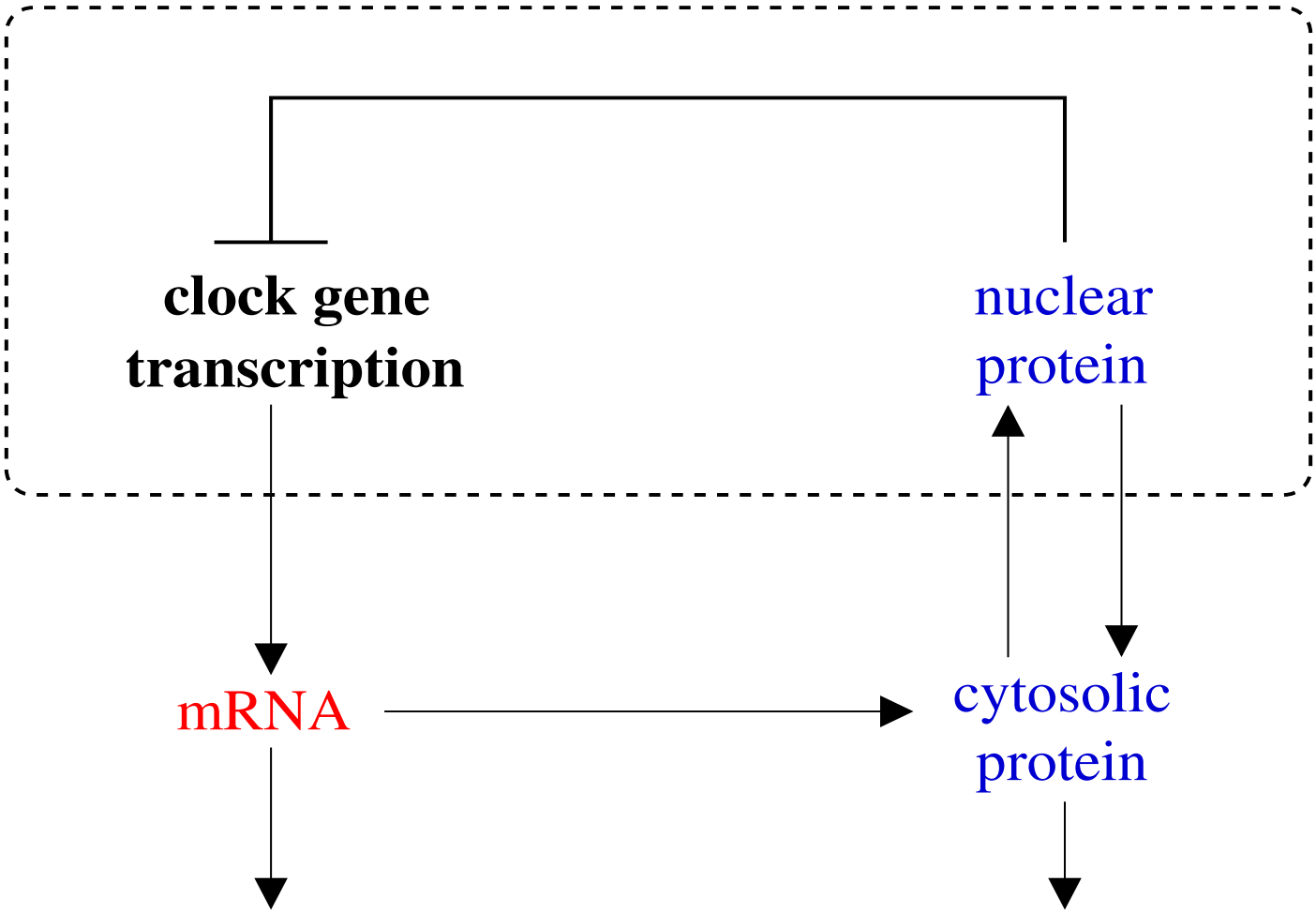
Reasoning

- Process algebras are equipped with equivalence relations, and partial relations.
- These allow reasoning about the relationships between models: either alternative representations (as we have seen) or models which result from simplification or elaboration of an original model.
- Additionally for some process algebras there are complementary modal logics which allow system properties to be formally expressed and automatically checked.

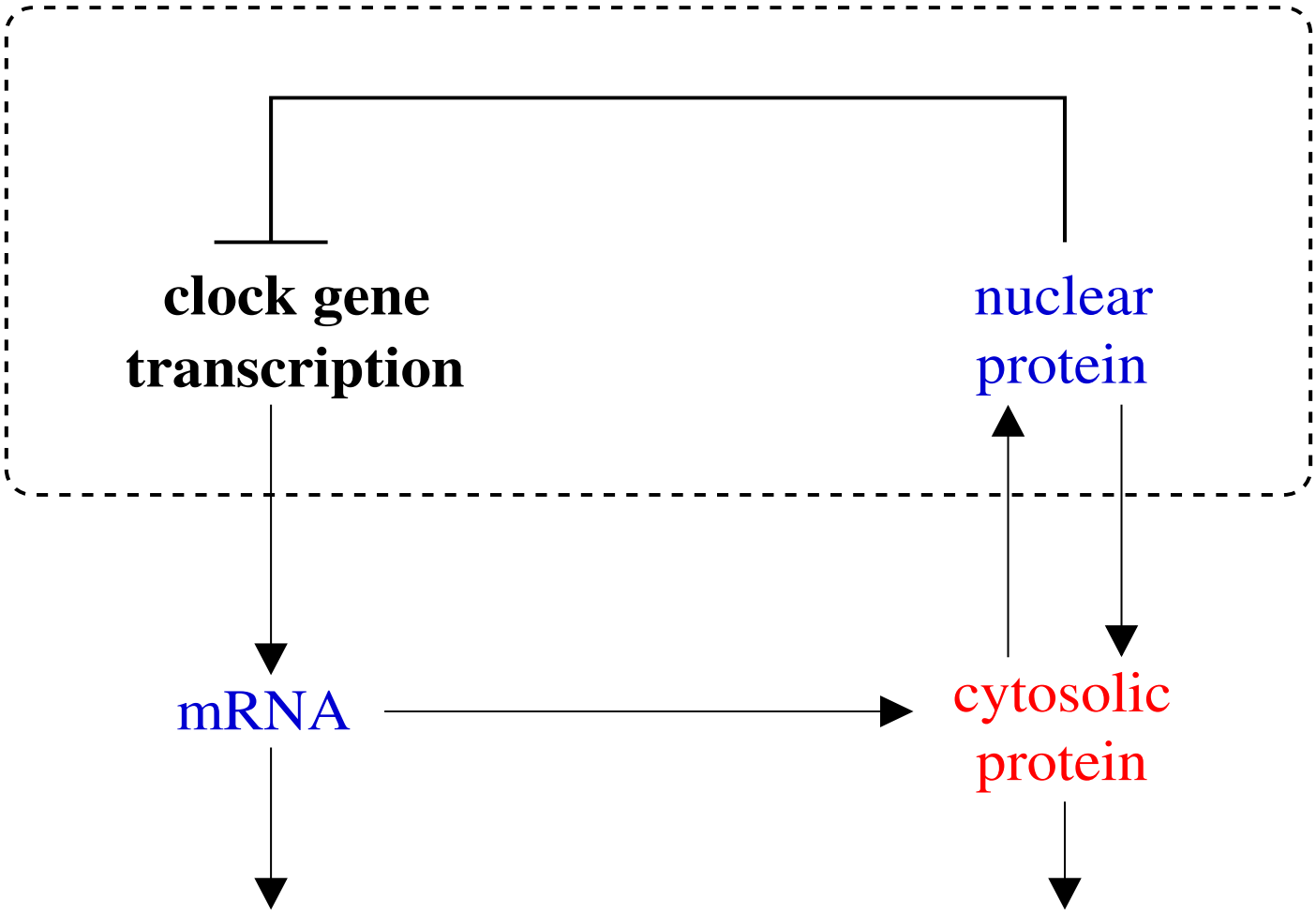
A simple circadian clock



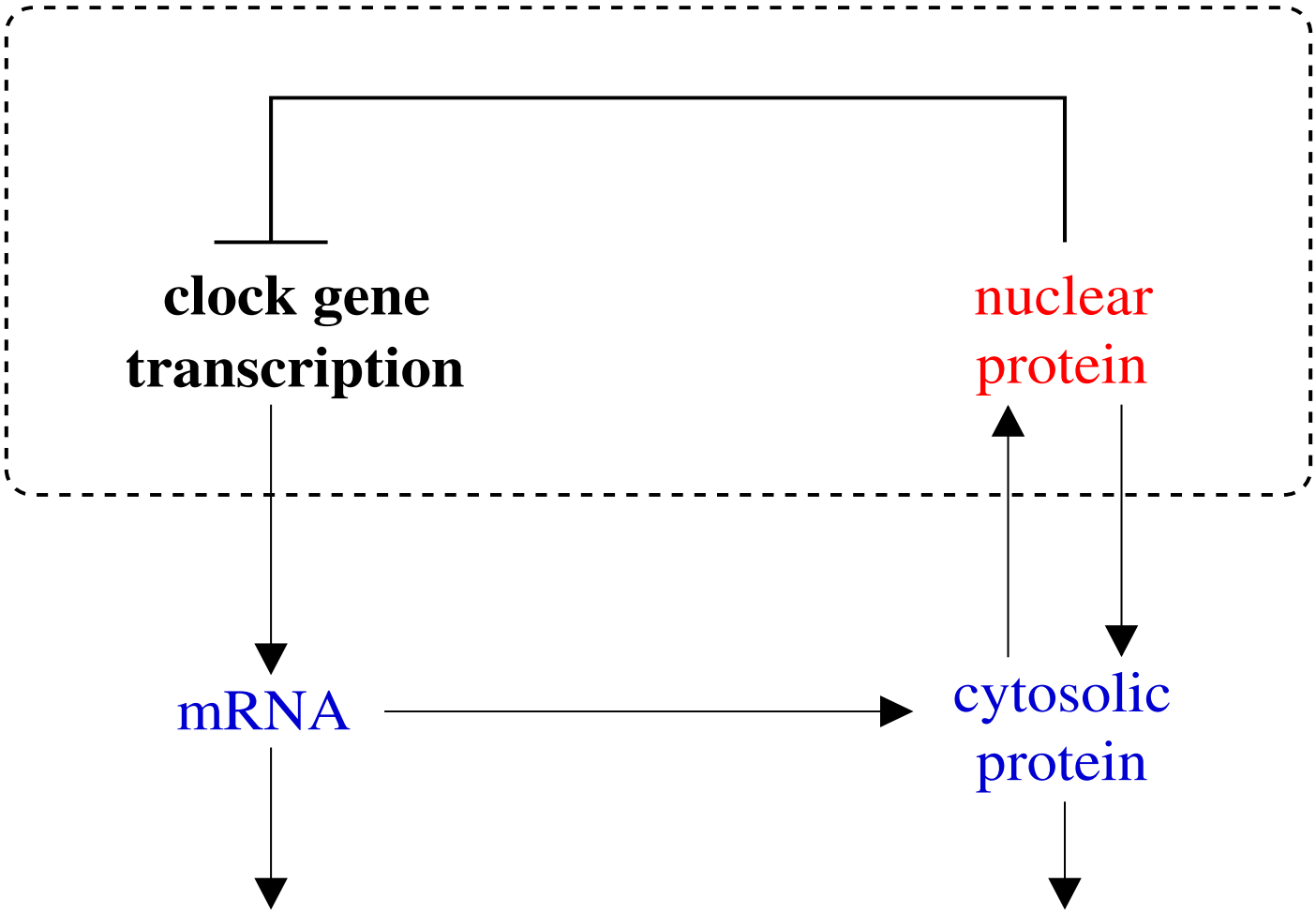
A simple circadian clock



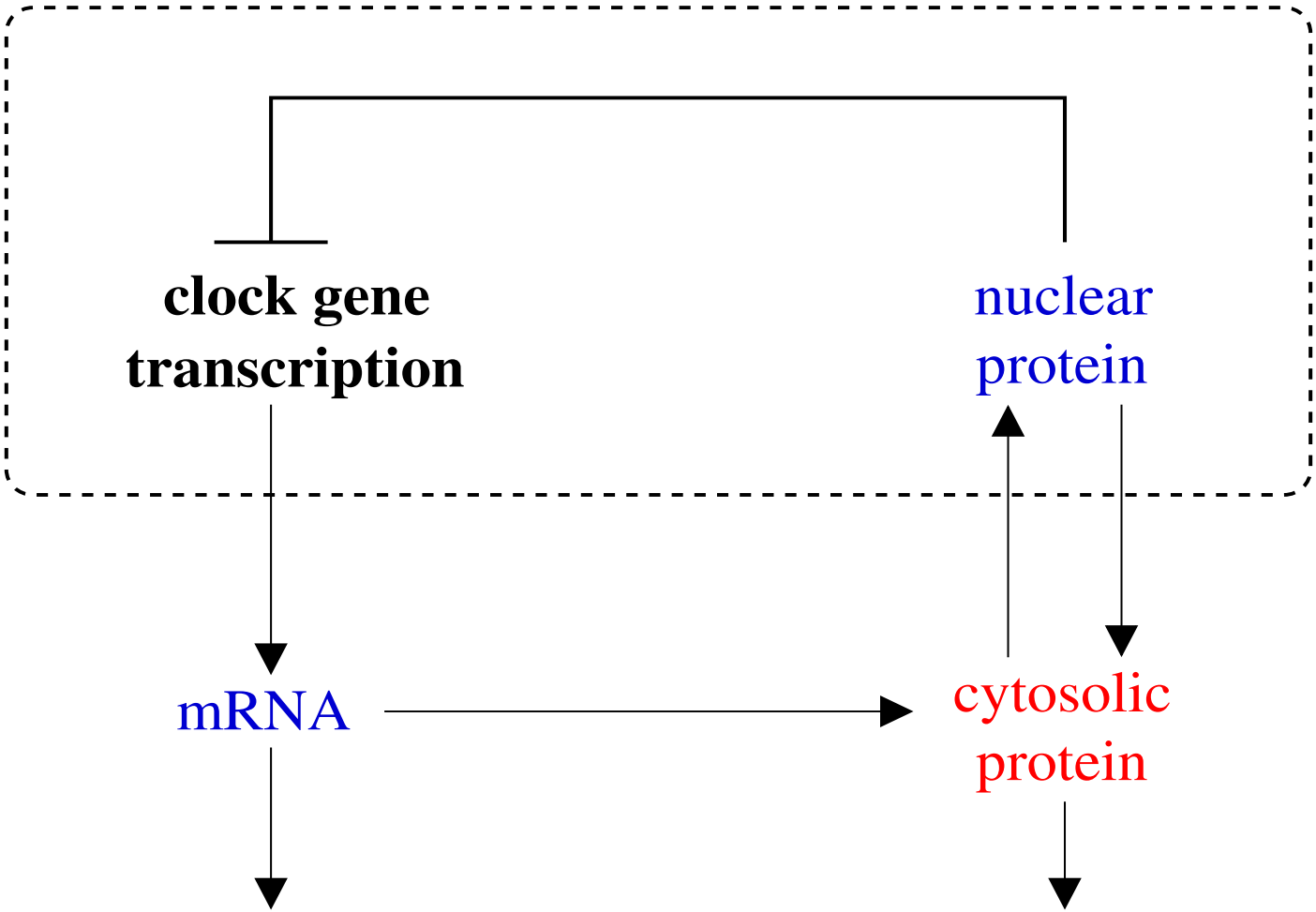
A simple circadian clock



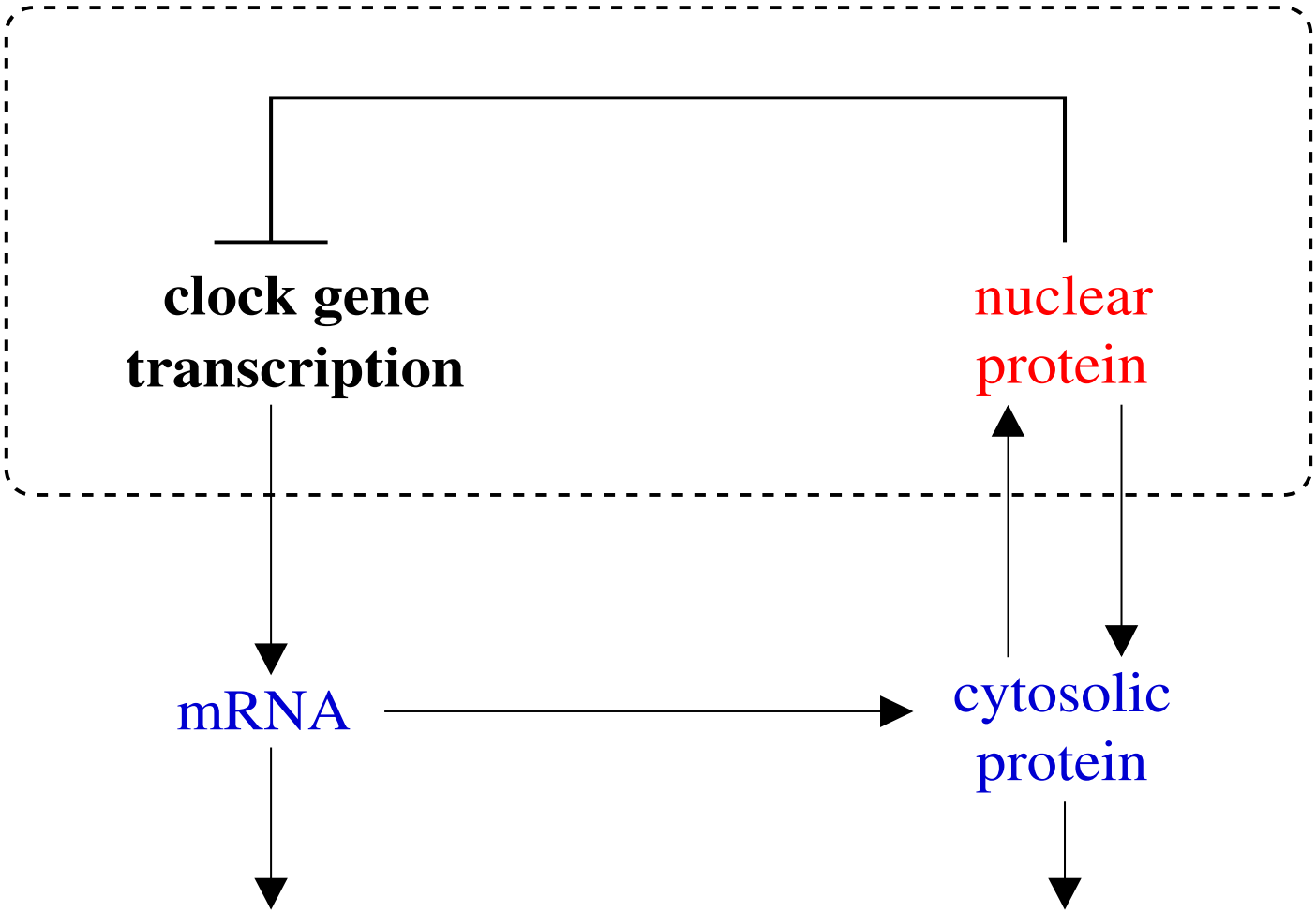
A simple circadian clock



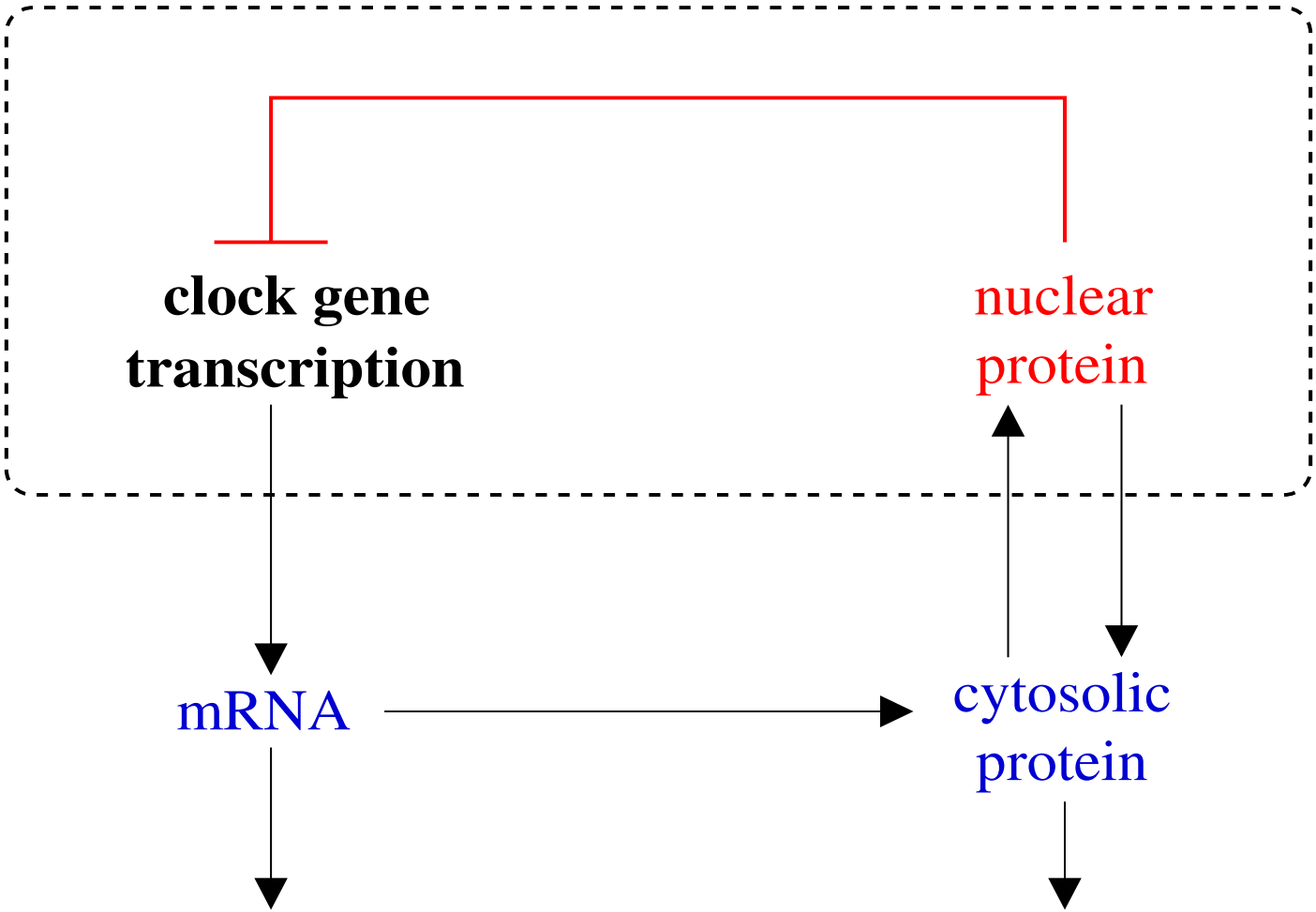
A simple circadian clock



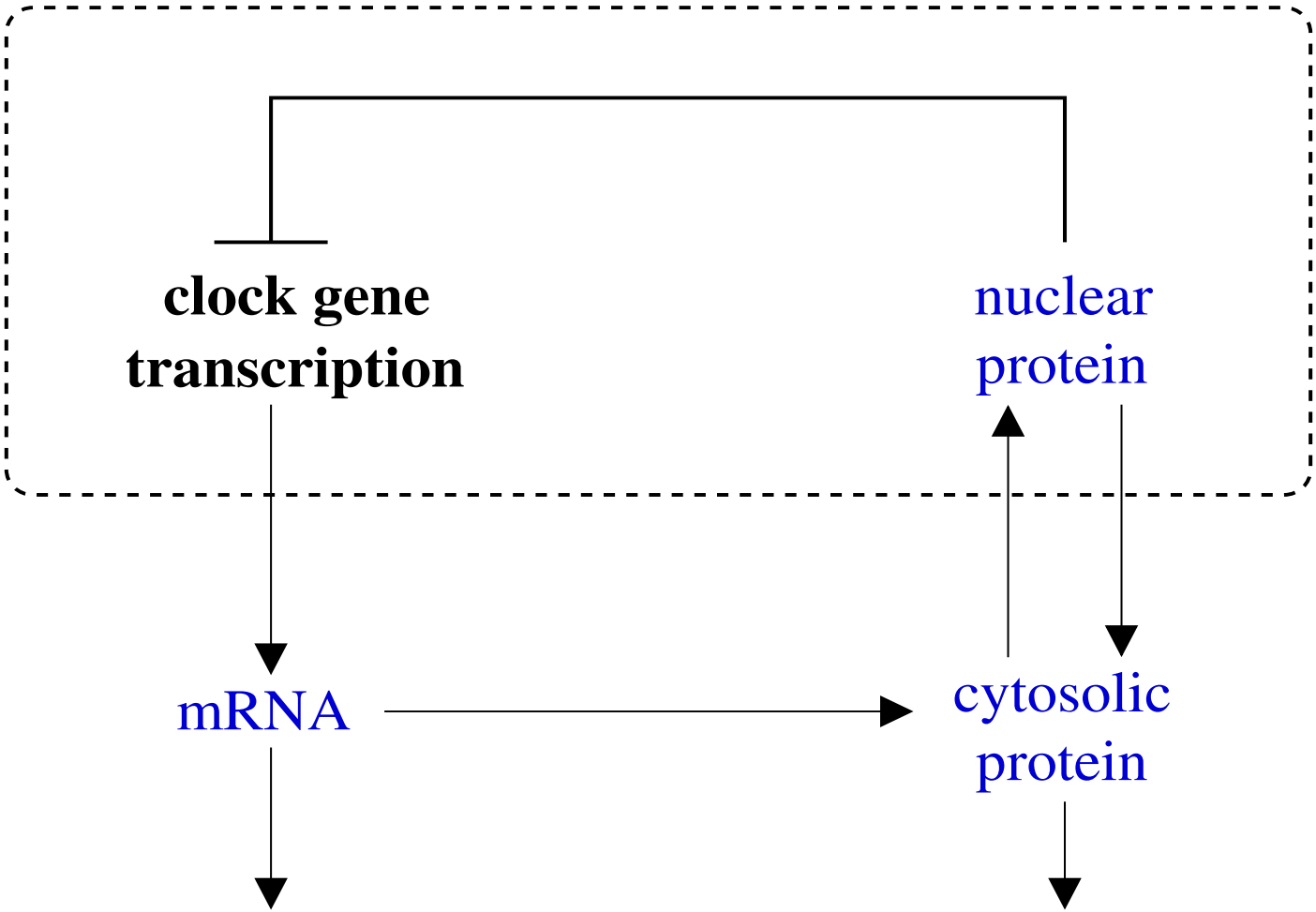
A simple circadian clock



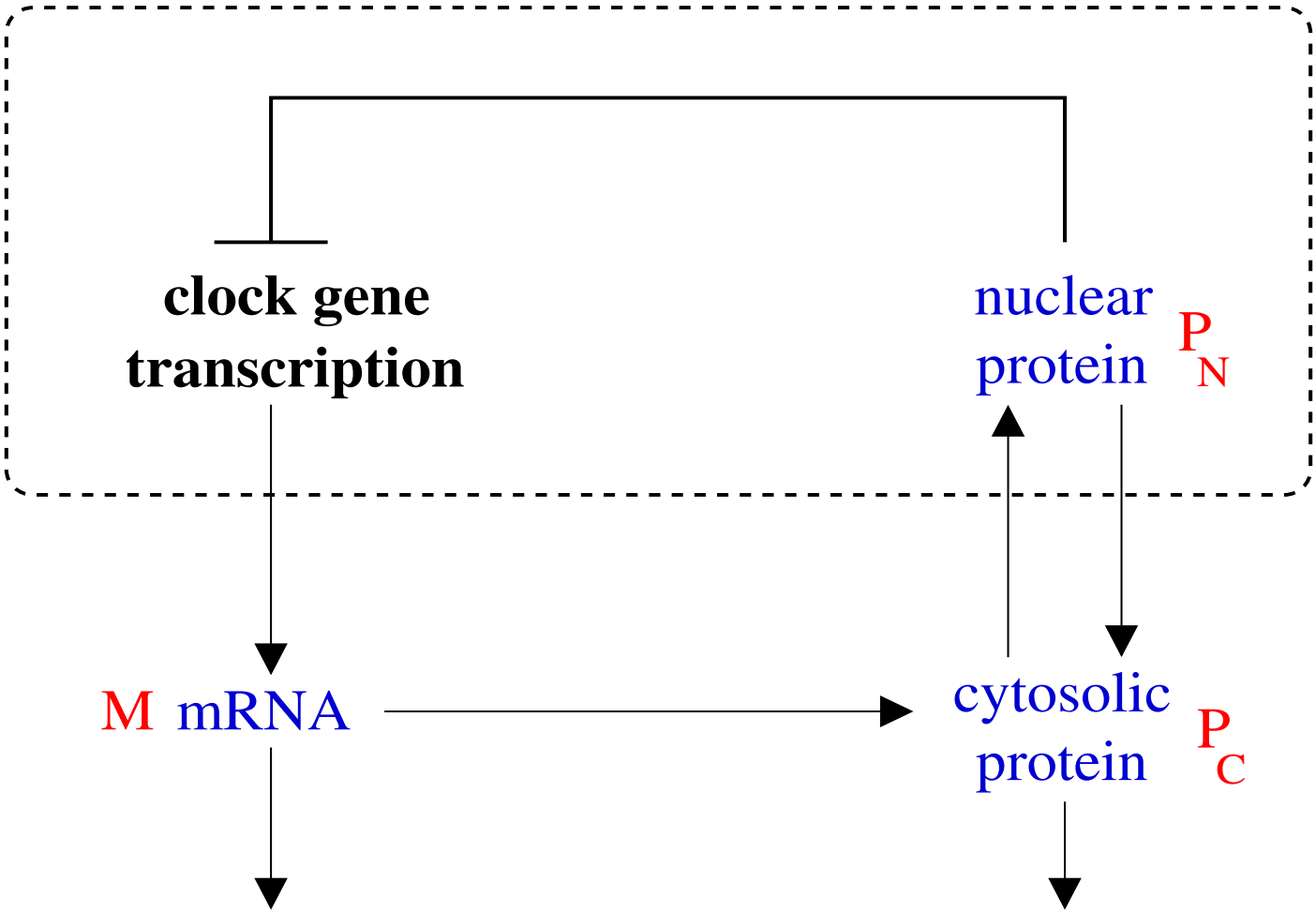
A simple circadian clock



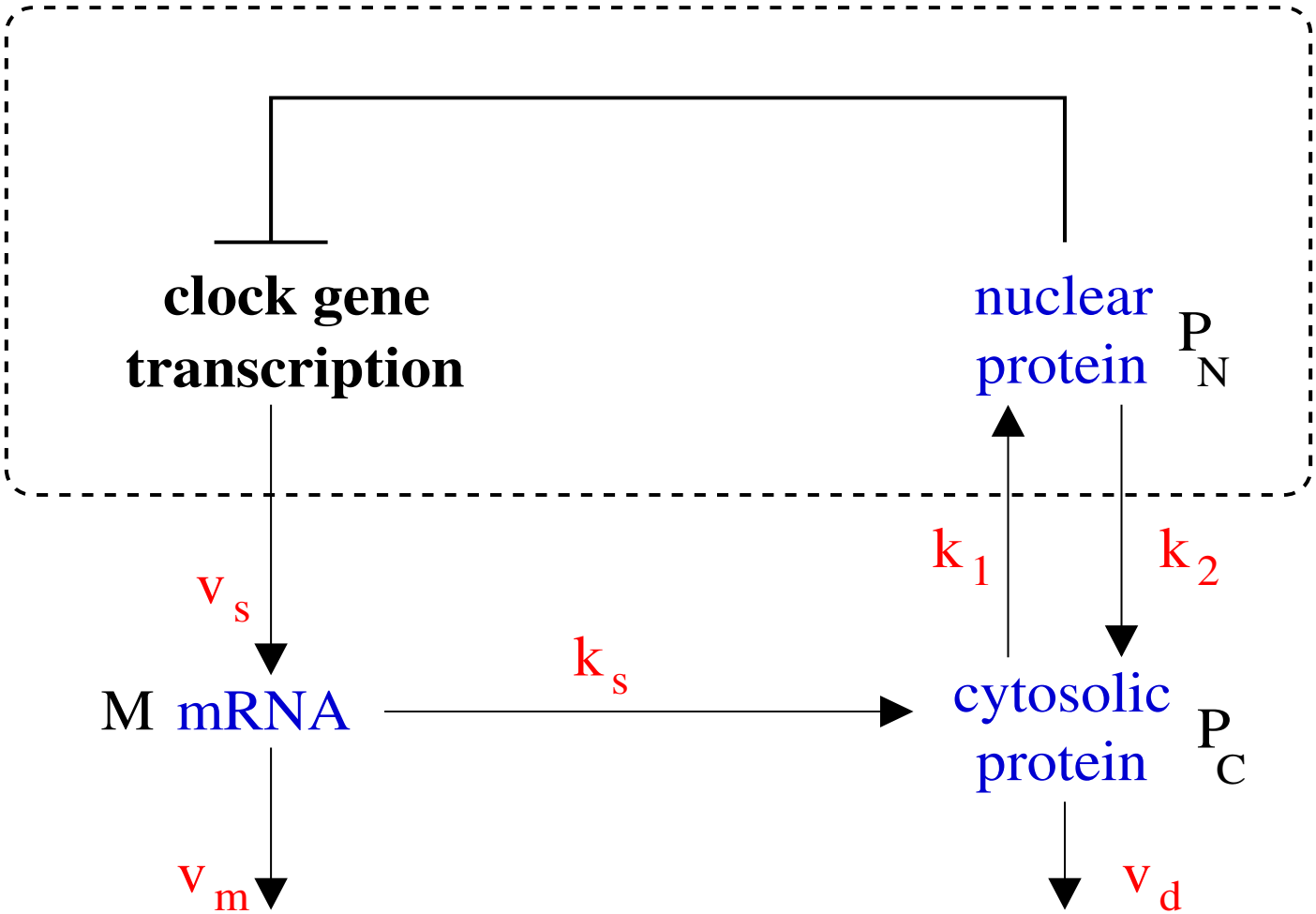
A simple circadian clock



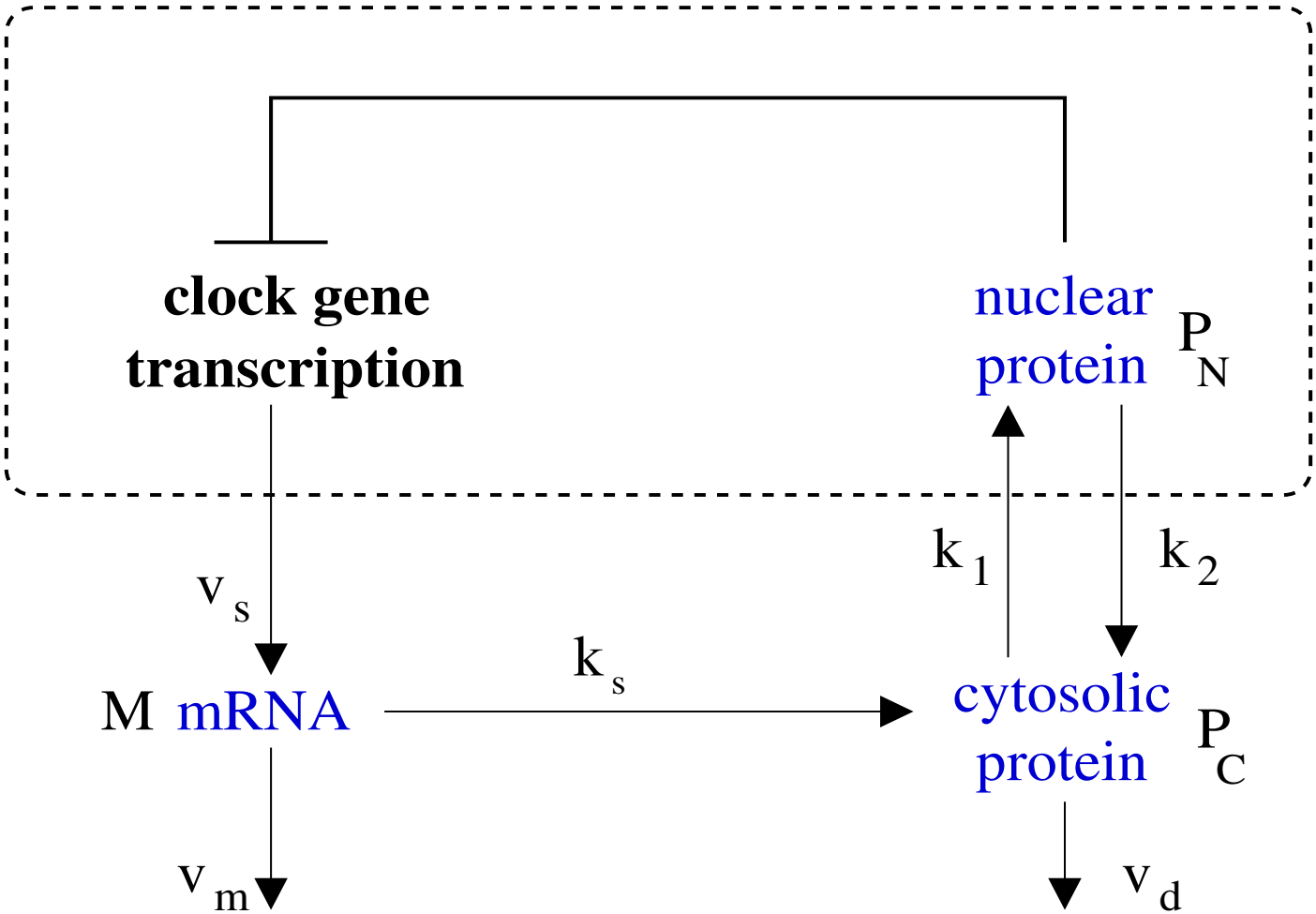
A simple circadian clock



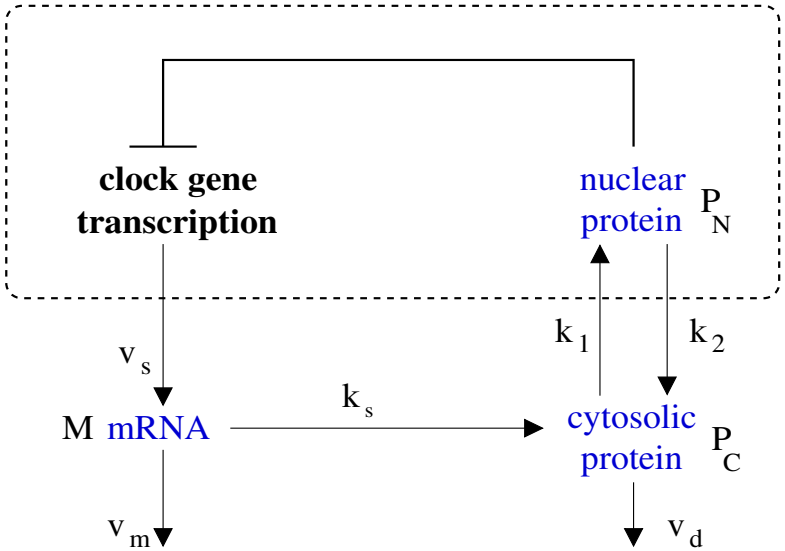
A simple circadian clock



A simple circadian clock



Handcrafted ODEs

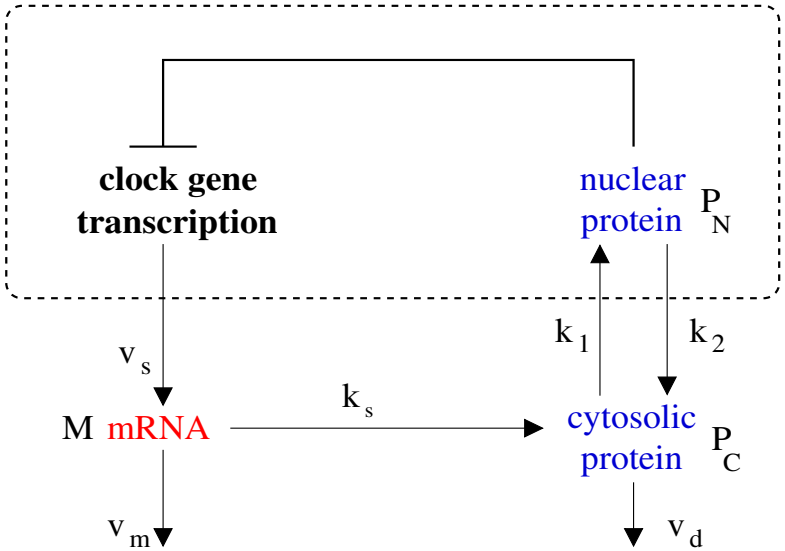


$$\frac{d[M]}{dt} = v_s \frac{k_I^n}{k_I^n + [P_N]^n} - v_m \frac{[M]}{k_m + [M]}$$

$$\frac{d[P_C]}{dt} = k_s[M] - v_d \frac{[P_C]}{k_d + [P_C]} - k_1[P_C] + k_2[P_N]$$

$$\frac{d[P_N]}{dt} = k_1[P_C] - k_2[P_N]$$

Handcrafted ODEs

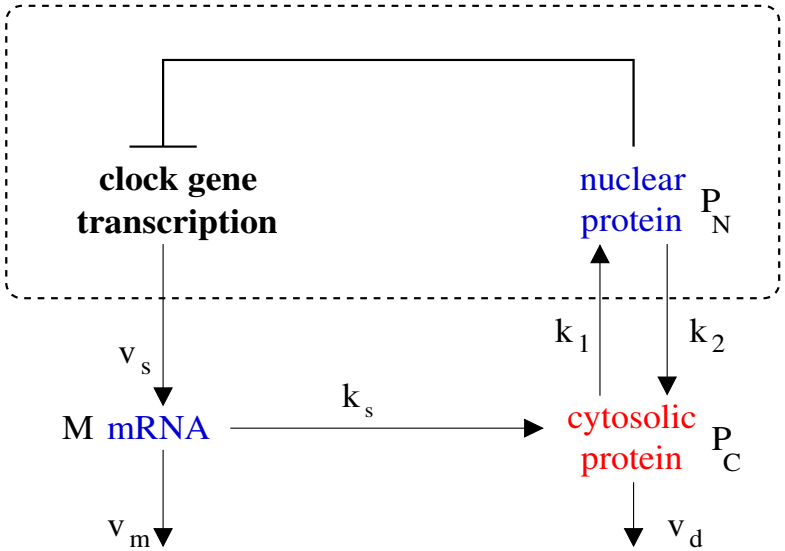


$$\frac{d[M]}{dt} = v_s \frac{k_I^n}{k_I^n + [P_N]^n} - v_m \frac{[M]}{k_m + [M]}$$

$$\frac{d[P_C]}{dt} = k_s[M] - v_d \frac{[P_C]}{k_d + [P_C]} - k_1[P_C] + k_2[P_N]$$

$$\frac{d[P_N]}{dt} = k_1[P_C] - k_2[P_N]$$

Handcrafted ODEs

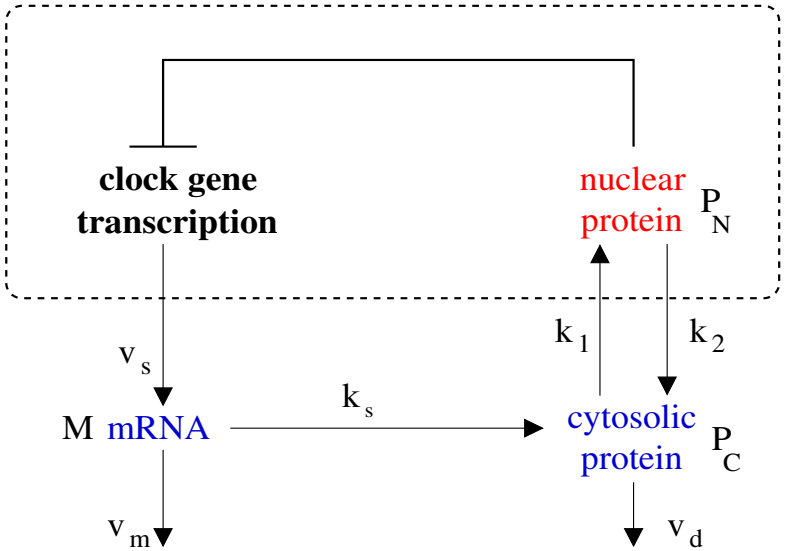


$$\frac{d[M]}{dt} = v_s \frac{k_I^n}{k_I^n + [P_N]^n} - v_m \frac{[M]}{k_m + [M]}$$

$$\frac{d[P_C]}{dt} = k_s[M] - v_d \frac{[P_C]}{k_d + [P_C]} - k_1[P_C] + k_2[P_N]$$

$$\frac{d[P_N]}{dt} = k_1[P_C] - k_2[P_N]$$

Handcrafted ODEs



$$\frac{d[M]}{dt} = v_s \frac{k_I^n}{k_I^n + [P_N]^n} - v_m \frac{[M]}{k_m + [M]}$$

$$\frac{d[P_C]}{dt} = k_s[M] - v_d \frac{[P_C]}{k_d + [P_C]} - k_1[P_C] + k_2[P_N]$$

$$\frac{d[P_N]}{dt} = k_1[P_C] - k_2[P_N]$$

Representing the circadian clock in PEPA

- Some of the "steps" in the biological representation (diagram) and the corresponding ODEs do not correspond to elementary reaction steps:

Representing the circadian clock in PEPA

- Some of the "steps" in the biological representation (diagram) and the corresponding ODEs do not correspond to elementary reaction steps:
  - To use our current mappings we need to decompose the enzyme-substrate and gene-repressor reactions into elementary steps.

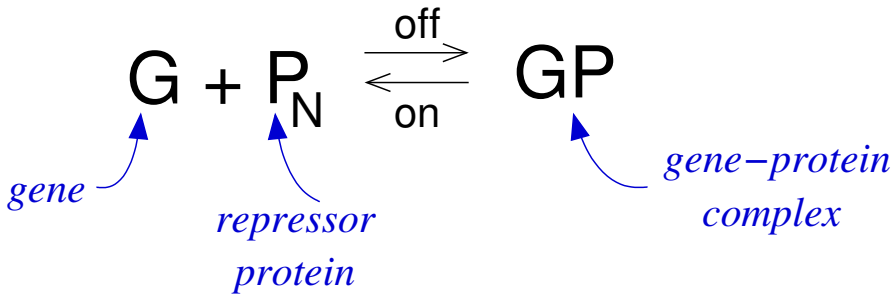
Representing the circadian clock in PEPA

- Some of the "steps" in the biological representation (diagram) and the corresponding ODEs do not correspond to elementary reaction steps:
  - To use our current mappings we need to decompose the enzyme-substrate and gene-repressor reactions into elementary steps.
- PEPA does not have combinators to express repression or catalysis:

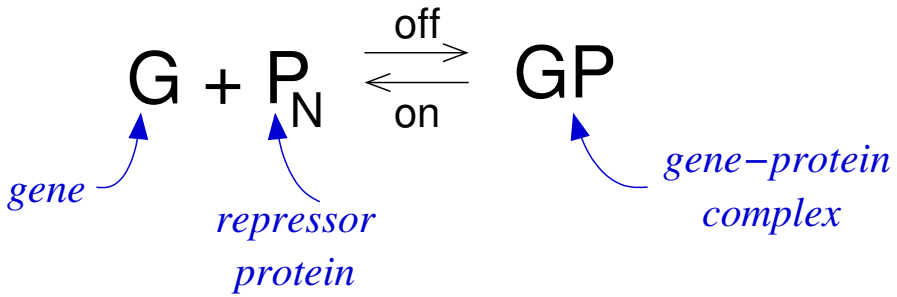
Representing the circadian clock in PEPA

- Some of the "steps" in the biological representation (diagram) and the corresponding ODEs do not correspond to elementary reaction steps:
  - To use our current mappings we need to decompose the enzyme-substrate and gene-repressor reactions into elementary steps.
- PEPA does not have combinators to express repression or catalysis:
  - We introduce additional abstract components to the PEPA model which do not correspond to species but to transcription and repression.

Repression mechanism



Repression mechanism

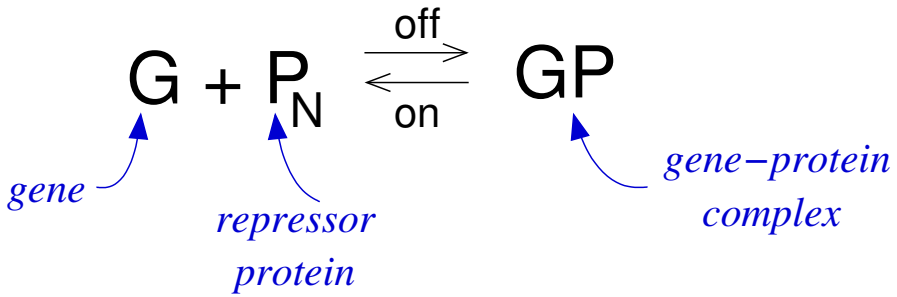


$$\text{Transcription} \begin{cases} T^h \stackrel{def}{=} (transcribe, v_s).T^h + (off, \top).T^l \\ T^l \stackrel{def}{=} (on, \top).T^h \end{cases}$$

$$\text{Repression} \begin{cases} R^h \stackrel{def}{=} (on, v_{on}).R^l \\ R^l \stackrel{def}{=} (off, \top).R^h \end{cases}$$

$$P_N^h \stackrel{def}{=} (off, v_{off}).P_N^l + \ldots$$

Repression mechanism



$$\text{Transcription} \begin{cases} T^h \stackrel{def}{=} (transcribe, v_s).T^h + (off, \top).T^l \\ T^l \stackrel{def}{=} (on, \top).T^h \end{cases}$$

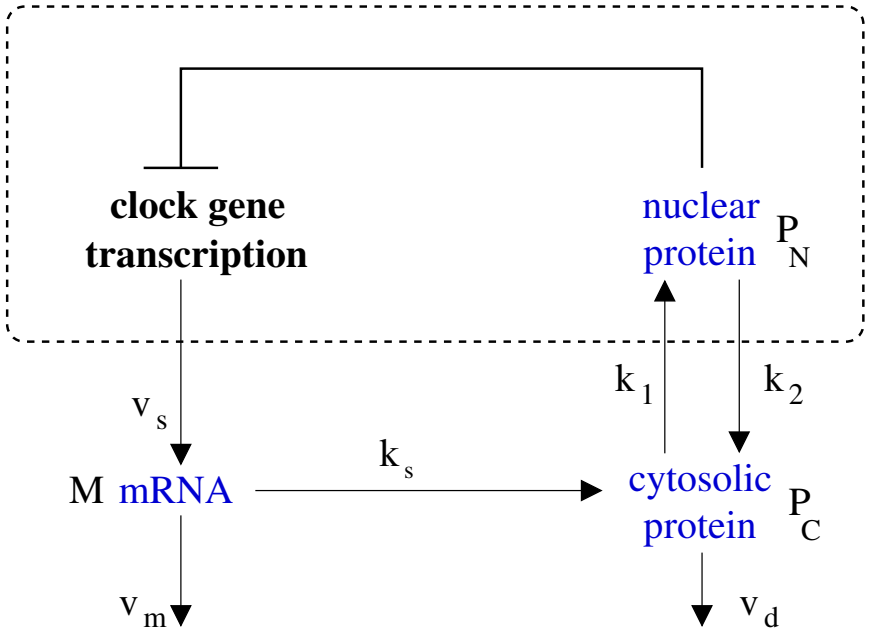
$$\text{Repression} \begin{cases} R^h \stackrel{def}{=} (on, v_{on}).R^l \\ R^l \stackrel{def}{=} (off, \top).R^h \end{cases}$$

$$P_N^h \stackrel{def}{=} (off, v_{off}).P_N^l + \ldots$$

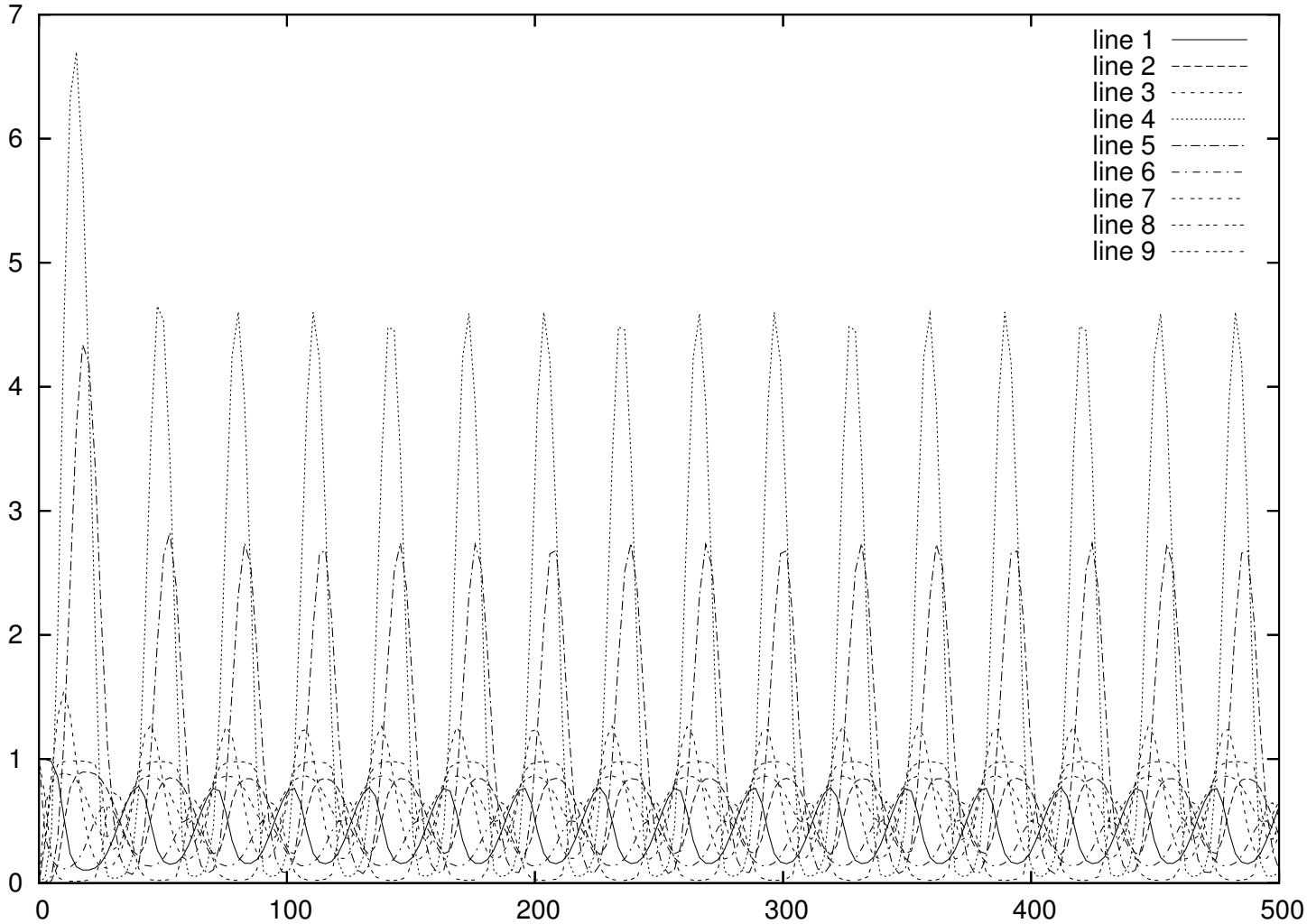
Only P_N is explicitly modelled; T and R are abstract entities.

PEPA model of the circadian clock

$$T^h \underset{\{transcribe,on,off\}}{\bowtie} \left(R^I \underset{\{off\}}{\bowtie} \left(\left(M \underset{\{translate\}}{\bowtie} (P_C \underset{\{trans_1,trans_2\}}{\bowtie} P_N) \right) \right) \right)$$



Results of quantitative analysis



Reasons to be cheerful

Previous work on PEPA in the performance modelling domain gives various reasons to be optimistic:

Reasons to be cheerful

Previous work on PEPA in the performance modelling domain gives various reasons to be optimistic:

- PEPA allowed rigorous development of the underlying mathematical models and formalised model manipulations and reductions;

Reasons to be cheerful

Previous work on PEPA in the performance modelling domain gives various reasons to be optimistic:

- PEPA allowed rigorous development of the underlying mathematical models and formalised model manipulations and reductions;
- Process algebras and other formal modelling techniques became integrated into performance modelling methodology, although sometimes embedded rather than on the surface (UML etc).

Reasons to be cheerful

Previous work on PEPA in the performance modelling domain gives various reasons to be optimistic:

- PEPA allowed rigorous development of the underlying mathematical models and formalised model manipulations and reductions;
- Process algebras and other formal modelling techniques became integrated into performance modelling methodology, although sometimes embedded rather than on the surface (UML etc).
- This work stimulated a lot of other work on formal approaches to performance modelling such as the development of suitably quantified modal logic and model checking.

Outline

Systems Biology: Will it Work?

Systems Biology: Will it Work?

Two quotes from Mesarović (1968):

Systems Biology: Will it Work?

Two quotes from Mesarović (1968):

The real advance in the application of systems theory to biology will come about only when the biologists start asking questions which are based on the systems-theoretic concepts rather than using these concepts to represent in still another way the phenomena which are already explained in terms of biophysical or biochemical principles.

Systems Biology: Will it Work?

Two quotes from Mesarović (1968):

The real advance in the application of systems theory to biology will come about only when the biologists start asking questions which are based on the systems-theoretic concepts rather than using these concepts to represent in still another way the phenomena which are already explained in terms of biophysical or biochemical principles.

The fundamental question for the community of biologists is whether an explanation on the systems theoretic basis is acceptable as a true scientific explanation of the biological inquiry.

What's life got to do with it?

What's life got to do with it?

"Life is a relationship among molecules and not a property of any molecule"

[Linus Pauling]

Thank you!