

From Simulative Programs *as* Theories to Theories *of* Simulative Programs

NICOLA ANGIUS ¹

(Work in conjunction with Guglielmo Tamburrini)

¹Department of History, Human Sciences, and Education.
University of Sassari, Italy
nangius@uniss.it

Paris. February, 23, 2017

Motivations

- The epistemological status of programs in simulative Artificial Intelligence (AI) and Artificial Life (AL).
- Methodological analysis of current simulative studies in computational biology: the case of Executable Cell Biology (ECB) (Fisher and Henzinger 2007).
- Imports from the Philosophy of Computer Science.

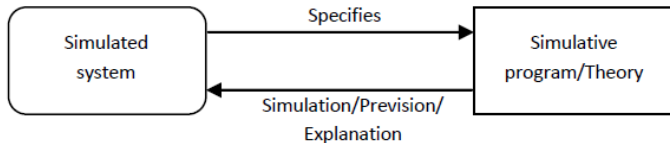
Motivations

- The epistemological status of programs in simulative Artificial Intelligence (AI) and Artificial Life (AL).
- Methodological analysis of current simulative studies in computational biology: the case of Executable Cell Biology (ECB) (Fisher and Henzinger 2007).
- Imports from the Philosophy of Computer Science.

Motivations

- The epistemological status of programs in simulative Artificial Intelligence (AI) and Artificial Life (AL).
- Methodological analysis of current simulative studies in computational biology: the case of Executable Cell Biology (ECB) (Fisher and Henzinger 2007).
- Imports from the Philosophy of Computer Science.

The simulative methodological approach



Human Problem Solving (Newell and Simon 1972)

1. A human agent is asked to solve a given problem (logical exercise or chess move) and to think out loud.
2. Verbal reports are analysed with the purpose of identifying the solution strategies of the agent.
3. The analysis of verbal reports is used to develop a computer program P that simulates the behaviour of the human agent.
4. Both the program and the human agent are asked to carry out a new problem solving task, and a comparison is made between the verbal reports of the agent and the program's execution traces.

Human Problem Solving (Newell and Simon 1972)

1. A human agent is asked to solve a given problem (logical exercise or chess move) and to think out loud.
2. Verbal reports are analysed with the purpose of identifying the solution strategies of the agent.
3. The analysis of verbal reports is used to develop a computer program P that simulates the behaviour of the human agent.
4. Both the program and the human agent are asked to carry out a new problem solving task, and a comparison is made between the verbal reports of the agent and the program's execution traces.

Human Problem Solving (Newell and Simon 1972)

1. A human agent is asked to solve a given problem (logical exercise or chess move) and to think out loud.
2. Verbal reports are analysed with the purpose of identifying the solution strategies of the agent.
3. The analysis of verbal reports is used to develop a computer program P that simulates the behaviour of the human agent.
4. Both the program and the human agent are asked to carry out a new problem solving task, and a comparison is made between the verbal reports of the agent and the program's execution traces.

Human Problem Solving (Newell and Simon 1972)

1. A human agent is asked to solve a given problem (logical exercise or chess move) and to think out loud.
2. Verbal reports are analysed with the purpose of identifying the solution strategies of the agent.
3. The analysis of verbal reports is used to develop a computer program P that simulates the behaviour of the human agent.
4. Both the program and the human agent are asked to carry out a new problem solving task, and a comparison is made between the verbal reports of the agent and the program's execution traces.

Simulative Programs as Theories

From a formal standpoint, a computer program used as a theory has the same epistemological status as a set of differential equations or difference equations used as a theory: (1) given a set of initial and boundary conditions, the differential equations predict the successive states of the system at subsequent points in time; (2) given a set of initial and subsequent environmental inputs, the computer program predicts the successive state of the system (the subject's symbol emissions and the state of his memory) at subsequent points in time (Newell e Simon 1960, p. 2013).

Simulative Programs as Theories

There is a well established list of advantages that programs bring to a theorist: they concentrate the mind marvelously; they transform mysticism into information processing, forcing the theorist to make intuitions explicit and to translate vague terminology into concrete proposals; they provide a secure test of the consistency of a theory and thereby allow complicated interactive components to be safely assembled; they are working models whose behavior can be directly compared with human performance.(Johnson-Laird, 1981, p. 185).

Simulative Programs as Theories

[T]he . . . requirement - that we be able to implement [a cognitive] process in terms of an actual, running program that exhibits tokens of the behaviors in question, under the appropriate circumstances - has farreaching consequences. One of the clearest advantages of expressing a cognitive-process model in the form of a computer program is, it provides a remarkable intellectual prosthetic for dealing with complexity and for exploring both the entailments of a large set of proposed principles and their interactions.(Pylyshyn, 1984, p. 76).

Three problems

1. A program-theory can incorporate, as a program, implementation details that are irrelevant for the processes to be simulated (Thagard 1984; Cooper and Guest 2014).
2. The limitations of the predictive and explanatory power of a simulative program with respect to the simulated system:
 - Calculation of the primitive recursive function $f(m, x, y)$ for a deterministic Turing Machine;
 - Limitations of Software Testing (Angius 2014; Symons and Horner 2014).
3. The problem of program correctness.

Three problems

1. A program-theory can incorporate, as a program, implementation details that are irrelevant for the processes to be simulated (Thagard 1984; Cooper and Guest 2014).
2. The limitations of the predictive and explanatory power of a simulative program with respect to the simulated system:
 - Calculation of the primitive recursive function $f(m, x, y)$ for a deterministic Turing Machine;
 - Limitations of Software Testing (Angius 2014; Symons and Horner 2014).
3. The problem of program correctness.

Three problems

1. A program-theory can incorporate, as a program, implementation details that are irrelevant for the processes to be simulated (Thagard 1984; Cooper and Guest 2014).
2. The limitations of the predictive and explanatory power of a simulative program with respect to the simulated system:
 - Calculation of the primitive recursive function $f(m, x, y)$ for a deterministic Turing Machine;
 - Limitations of Software Testing (Angius 2014; Symons and Horner 2014).
3. The problem of program correctness.

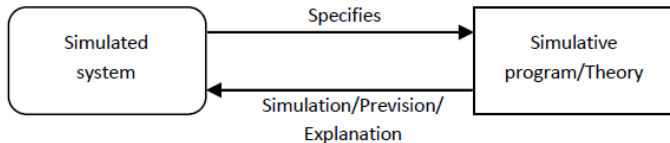
Three problems

1. A program-theory can incorporate, as a program, implementation details that are irrelevant for the processes to be simulated (Thagard 1984; Cooper and Guest 2014).
2. The limitations of the predictive and explanatory power of a simulative program with respect to the simulated system:
 - Calculation of the primitive recursive function $f(m, x, y)$ for a deterministic Turing Machine;
 - Limitations of Software Testing (Angius 2014; Symons and Horner 2014).
3. The problem of program correctness.

Three problems

1. A program-theory can incorporate, as a program, implementation details that are irrelevant for the processes to be simulated (Thagard 1984; Cooper and Guest 2014).
2. The limitations of the predictive and explanatory power of a simulative program with respect to the simulated system:
 - Calculation of the primitive recursive function $f(m, x, y)$ for a deterministic Turing Machine;
 - Limitations of Software Testing (Angius 2014; Symons and Horner 2014).
3. The problem of program correctness.

The simulative methodological approach



Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell sub-systems are represented by means of dynamical systems (quantitative models).

Simulative programs are built to compute the solutions of the differential equations involved in the dynamical systems thereby mimicking the evolution of the modelled cell system.

Difficulties:

1. Calculating the equations' solutions;
2. Specifying all parameters;
3. Evaluation of qualitative temporal properties.

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Cell systems are not input-output machines:

1. Behaviours depend on rates, positioning, and concurrences of received stimuli;
2. Are known for their homeostatic properties and their abilities of reacting to environmental modifications to preserve equilibrium.

Cell systems as *Reactive Systems*:

1. Are characterized by never-ending computations modelling cell systems' robustness and resilience;
2. Are concurrent systems obtained by the parallel composition of many computational processes;
3. Reactive systems can be examined by the Model Checking technique (Baier and Katoen 2008).

Biological networks: reaction networks - regulatory networks

- ▶ A biological network is modelled as a state transition system S ;
- ▶ A qualitative property is formalized using a temporal logic formula f ;
- ▶ Model checking is applied to verify whether $S \models f$.

Biological networks: reaction networks - regulatory networks

- ▶ A biological network is modelled as a state transition system S ;
- ▶ A qualitative property is formalized using a temporal logic formula f ;
- ▶ Model checking is applied to verify whether $S \models f$.

Biological networks: reaction networks - regulatory networks

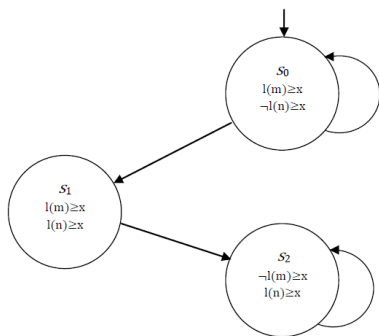
- ▶ A biological network is modelled as a state transition system S ;
- ▶ A qualitative property is formalized using a temporal logic formula f ;
- ▶ Model checking is applied to verify whether $S \models f$.

Biological networks: reaction networks - regulatory networks

- ▶ A biological network is modelled as a state transition system S ;
- ▶ A qualitative property is formalized using a temporal logic formula f ;
- ▶ Model checking is applied to verify whether $S \models f$.

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $F(\neg l(m) \geq x)$

Stability: $G(l(m) \geq x)$

Temporal ordering of events:
 $(l(m) \geq x)U(l(n) \geq x)$

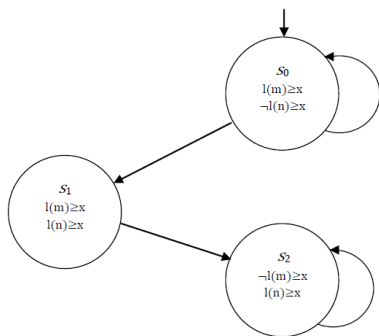
Correlation of concentrations:
 $G(l(m) \geq x) \Rightarrow F(l(n) \geq x)$

$M \models F(\neg l(m) \geq x)$; $M \models (l(m) \geq x)U(l(n) \geq x)$ \rightarrow WITNESSES

$M \not\models G(l(m) \geq x)$; $M \not\models G(l(m) \geq x) \Rightarrow F(l(n) \geq x)$ \rightarrow COUNTEREXAMPLES

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg l(m) \geq x)$

Stability: $\mathbf{G}(l(m) \geq x)$

Temporal ordering of events:
 $(l(m) \geq x)\mathbf{U}(l(n) \geq x)$

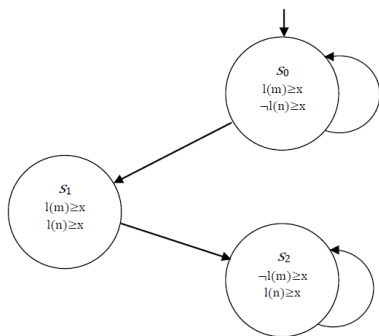
Correlation of concentrations:
 $\mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$

$M \models \mathbf{F}(\neg l(m) \geq x)$; $M \models (l(m) \geq x)\mathbf{U}(l(n) \geq x)$ → WITNESSES

$M \not\models \mathbf{G}(l(m) \geq x)$; $M \not\models \mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$ → COUNTEREXAMPLES

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg l(m) \geq x)$

Stability: $\mathbf{G}(l(m) \geq x)$

Temporal ordering of events:
 $(l(m) \geq x) \mathbf{U} (l(n) \geq x)$

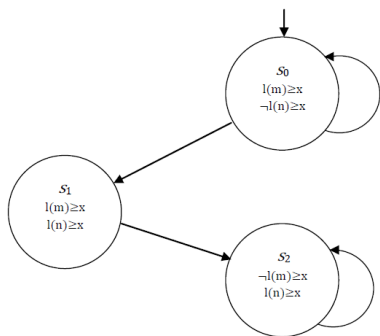
Correlation of concentrations:
 $\mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$

$M \models \mathbf{F}(\neg l(m) \geq x)$; $M \models (l(m) \geq x) \mathbf{U} (l(n) \geq x)$ \rightarrow WITNESSES

$M \not\models \mathbf{G}(l(m) \geq x)$; $M \not\models \mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$ \rightarrow COUNTEREXAMPLES

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg I(m) \geq x)$

Stability: $\mathbf{G}(I(m) \geq x)$

Temporal ordering of events:
 $(I(m) \geq x)\mathbf{U}(I(n) \geq x)$

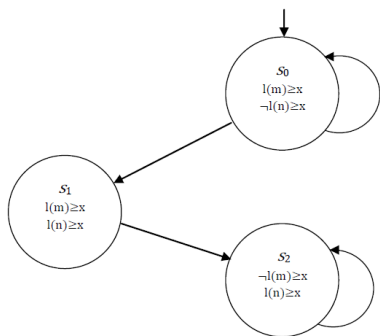
Correlation of concentrations:
 $\mathbf{G}(I(m) \geq x) \Rightarrow \mathbf{F}(I(n) \geq x)$

$M \models \mathbf{F}(\neg I(m) \geq x)$; $M \models (I(m) \geq x)\mathbf{U}(I(n) \geq x)$ \rightarrow WITNESSES

$M \not\models \mathbf{G}(I(m) \geq x)$; $M \not\models \mathbf{G}(I(m) \geq x) \Rightarrow \mathbf{F}(I(n) \geq x)$ \rightarrow COUNTEREXAMPLES

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg I(m) \geq x)$

Stability: $\mathbf{G}(I(m) \geq x)$

Temporal ordering of events:
 $(I(m) \geq x)\mathbf{U}(I(n) \geq x)$

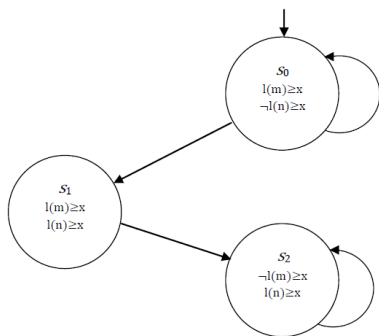
Correlation of concentrations:
 $\mathbf{G}(I(m) \geq x) \Rightarrow \mathbf{F}(I(n) \geq x)$

$M \models \mathbf{F}(\neg I(m) \geq x)$; $M \models (I(m) \geq x)\mathbf{U}(I(n) \geq x)$ \rightarrow WITNESSES

$M \not\models \mathbf{G}(I(m) \geq x)$; $M \not\models \mathbf{G}(I(m) \geq x) \Rightarrow \mathbf{F}(I(n) \geq x)$ \rightarrow COUNTEREXAMPLES

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg l(m) \geq x)$

Stability: $\mathbf{G}(l(m) \geq x)$

Temporal ordering of events:
 $(l(m) \geq x)\mathbf{U}(l(n) \geq x)$

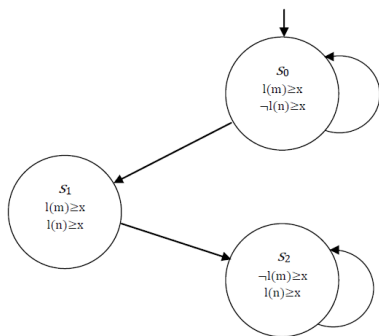
Correlation of concentrations:
 $\mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$

$M \models \mathbf{F}(\neg l(m) \geq x); M \models (l(m) \geq x)\mathbf{U}(l(n) \geq x) \quad \rightarrow \text{WITNESSES}$

$M \not\models \mathbf{G}(l(m) \geq x); M \not\models \mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x) \quad \rightarrow \text{COUNTEREXAMPLES}$

Example

Kripke Structure $M = (S, S_0, R, L)$



Temporal logic formulas

Reachability: $\mathbf{F}(\neg l(m) \geq x)$

Stability: $\mathbf{G}(l(m) \geq x)$

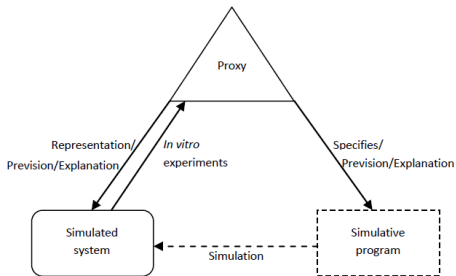
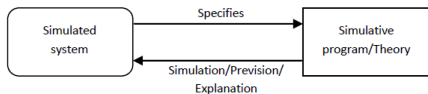
Temporal ordering of events:
 $(l(m) \geq x)\mathbf{U}(l(n) \geq x)$

Correlation of concentrations:
 $\mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$

$M \models \mathbf{F}(\neg l(m) \geq x)$; $M \models (l(m) \geq x)\mathbf{U}(l(n) \geq x)$ \rightarrow WITNESSES

$M \not\models \mathbf{G}(l(m) \geq x)$; $M \not\models \mathbf{G}(l(m) \geq x) \Rightarrow \mathbf{F}(l(n) \geq x)$ \rightarrow COUNTEREXAMPLES

Triangulation of simulative method in ECB



The simulative method in ECB

1. The proxy provides a *system specification* for all the simulative programs of the biological network.

Circumscribes the class of all eligible simulative executions of the biological network, *abstracting from the specific ways of realizing a permissible execution with a given simulative program.*

2. The use of a proxy in the context of ECB extends the predictive power of simulative programs (limits of software testing).
3. Provides a means by which to prove correctness of simulative programs.

The simulative method in ECB

1. The proxy provides a *system specification* for all the simulative programs of the biological network.

Circumscribes the class of all eligible simulative executions of the biological network, *abstracting from the specific ways of realizing a permissible execution with a given simulative program.*

2. The use of a proxy in the context of ECB extends the predictive power of simulative programs (limits of software testing).
3. Provides a means by which to prove correctness of simulative programs.

The simulative method in ECB

1. The proxy provides a *system specification* for all the simulative programs of the biological network.

Circumscribes the class of all eligible simulative executions of the biological network, *abstracting from the specific ways of realizing a permissible execution with a given simulative program.*

2. The use of a proxy in the context of ECB extends the predictive power of simulative programs (limits of software testing).
3. Provides a means by which to prove correctness of simulative programs.

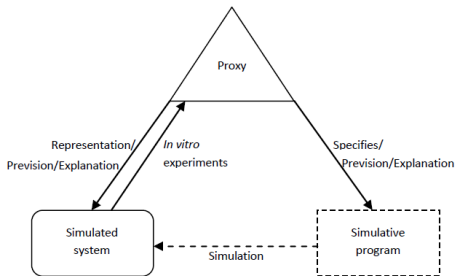
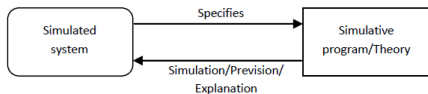
The simulative method in ECB

1. The proxy provides a *system specification* for all the simulative programs of the biological network.

Circumscribes the class of all eligible simulative executions of the biological network, *abstracting from the specific ways of realizing a permissible execution with a given simulative program.*

2. The use of a proxy in the context of ECB extends the predictive power of simulative programs (limits of software testing).
3. Provides a means by which to prove correctness of simulative programs.

Triangulation of simulative method in ECB



Corroboration and falsification of hypotheses in simulative AI

1. Empirical hypotheses are advanced directly on the natural system on the basis of initial observation.
2. Those hypotheses are used as a blueprint to build an artificial system (program or robot).
3. Behaviours of the artificial system are compared with some behaviours of interest of the natural system.
4. The hypotheses one started from are corroborated (or falsified) in case the behaviours of the artificial system match (or mismatch) with the behaviours of the natural system.

Corroboration and falsification of hypotheses in simulative AI

1. Empirical hypotheses are advanced directly on the natural system on the basis of initial observation.
2. Those hypotheses are used as a blueprint to build an artificial system (program or robot).
3. Behaviours of the artificial system are compared with some behaviours of interest of the natural system.
4. The hypotheses one started from are corroborated (or falsified) in case the behaviours of the artificial system match (or mismatch) with the behaviours of the natural system.

Corroboration and falsification of hypotheses in simulative AI

1. Empirical hypotheses are advanced directly on the natural system on the basis of initial observation.
2. Those hypotheses are used as a blueprint to build an artificial system (program or robot).
3. Behaviours of the artificial system are compared with some behaviours of interest of the natural system.
4. The hypotheses one started from are corroborated (or falsified) in case the behaviours of the artificial system match (or mismatch) with the behaviours of the natural system.

Corroboration and falsification of hypotheses in simulative AI

1. Empirical hypotheses are advanced directly on the natural system on the basis of initial observation.
2. Those hypotheses are used as a blueprint to build an artificial system (program or robot).
3. Behaviours of the artificial system are compared with some behaviours of interest of the natural system.
4. The hypotheses one started from are corroborated (or falsified) in case the behaviours of the artificial system match (or mismatch) with the behaviours of the natural system.

Corroboration and falsification of hypotheses in ECB

1. An initial set of property specifications is defined on the basis of data collected during wet experiments;
2. the cell system is described in terms of a Boolean representation instantiating those requirements;
3. a Kripke structure is extracted from the Boolean model so that all the potential ordering among allowed transitions are included.

Kripke structures are abductive *hypotheses* (Magnani et al 1999) on the modelled cell system's behaviours

Corroboration and falsification of hypotheses in ECB

1. An initial set of property specifications is defined on the basis of data collected during wet experiments;
2. the cell system is described in terms of a Boolean representation instantiating those requirements;
3. a Kripke structure is extracted from the Boolean model so that all the potential ordering among allowed transitions are included.

Kripke structures are abductive *hypotheses* (Magnani et al 1999) on the modelled cell system's behaviours

Corroboration and falsification of hypotheses in ECB

1. An initial set of property specifications is defined on the basis of data collected during wet experiments;
2. the cell system is described in terms of a Boolean representation instantiating those requirements;
3. a Kripke structure is extracted from the Boolean model so that all the potential ordering among allowed transitions are included.

Kripke structures are abductive *hypotheses* (Magnani et al 1999) on the modelled cell system's behaviours

Corroboration and falsification of hypotheses in ECB

1. An initial set of property specifications is defined on the basis of data collected during wet experiments;
2. the cell system is described in terms of a Boolean representation instantiating those requirements;
3. a Kripke structure is extracted from the Boolean model so that all the potential ordering among allowed transitions are included.

Kripke structures are abductive *hypotheses* (Magnani et al 1999) on the modelled cell system's behaviours

Corroboration and falsification of hypotheses in ECB

The hypothesis-model is *corroborated* by evaluating the empirical adequacy of the formal model:

1. The Kripke structure is model checked against the specifications that were advanced on the basis of wet experiments;
2. In case of negative answer of the model checking algorithm, the initial hypothesis (the model) is falsified and counterexamples are used to revise the hypothesis-model;
3. The model is checked against the remaining specifications and, if it resists falsification, the modified hypothesis is corroborated.

Corroboration and falsification of hypotheses in ECB

The hypothesis-model is *corroborated* by evaluating the empirical adequacy of the formal model:

1. The Kripke structure is model checked against the specifications that were advanced on the basis of wet experiments;
2. In case of negative answer of the model checking algorithm, the initial hypothesis (the model) is falsified and counterexamples are used to revise the hypothesis-model;
3. The model is checked against the remaining specifications and, if it resists falsification, the modified hypothesis is corroborated.

Corroboration and falsification of hypotheses in ECB

The hypothesis-model is *corroborated* by evaluating the empirical adequacy of the formal model:

1. The Kripke structure is model checked against the specifications that were advanced on the basis of wet experiments;
2. In case of negative answer of the model checking algorithm, the initial hypothesis (the model) is falsified and counterexamples are used to revise the hypothesis-model;
3. The model is checked against the remaining specifications and, if it resists falsification, the modified hypothesis is corroborated.

Corroboration and falsification of hypotheses in ECB

The hypothesis-model is *corroborated* by evaluating the empirical adequacy of the formal model:

1. The Kripke structure is model checked against the specifications that were advanced on the basis of wet experiments;
2. In case of negative answer of the model checking algorithm, the initial hypothesis (the model) is falsified and counterexamples are used to revise the hypothesis-model;
3. The model is checked against the remaining specifications and, if it resists falsification, the modified hypothesis is corroborated.

Discovering new regular behaviours in ECB

1. New hypotheses on the behaviours of the Kripke structure are advanced in terms of temporal logic formulas;
2. The model checking procedure evaluates whether those formula hold of the model;
3. In case of positive answer, witnesses are used to perform wet experiments to confirm the model-based nypotheses;
4. In case of negative answer, counterexamples are used as "coverage criteria" to perform wet experiments and decied whether the property specifications or the system specifications need to be revised.

Discovering new regular behaviours in ECB

1. New hypotheses on the behaviours of the Kripke structure are advanced in terms of temporal logic formulas;
2. The model checking procedure evaluates whether those formula hold of the model;
3. In case of positive answer, witnesses are used to perform wet experiments to confirm the model-based nypotheses;
4. In case of negative answer, counterexamples are used as "coverage criteria" to perform wet experiments and decied whether the property specifications or the system specifications need to be revised.

Discovering new regular behaviours in ECB

1. New hypotheses on the behaviours of the Kripke structure are advanced in terms of temporal logic formulas;
2. The model checking procedure evaluates whether those formula hold of the model;
3. In case of positive answer, witnesses are used to perform wet experiments to confirm the model-based nypotheses;
4. In case of negative answer, counterexamples are used as "coverage criteria" to perform wet experiments and decied whether the property specifications or the system specifications need to be revised.

Discovering new regular behaviours in ECB

1. New hypotheses on the behaviours of the Kripke structure are advanced in terms of temporal logic formulas;
2. The model checking procedure evaluates whether those formula hold of the model;
3. In case of positive answer, witnesses are used to perform wet experiments to confirm the model-based nypotheses;
4. In case of negative answer, counterexamples are used as "coverage criteria" to perform wet experiments and decied whether the property specifications or the system specifications need to be revised.

Conclusions

- ▶ ECB resumes the general idea of constructing theoretical models that are also executable, pursued over fifty years ago by Newell and Simon.

However, instead of a simulative program, ECB focuses on a more abstract model of the biological system:

1. the processes of abstraction omits from the model those implementation details of the simulative program that have no theoretical value;
 2. the executability of the abstract model permits to expand the class of predictions that can be extracted from the observation of the simulative programs executions.
- ▶ ECB modifies the discovery and corroboration processes of hypotheses on simulated systems in the methodology of simulative AI and AL.

Conclusions

- ▶ ECB resumes the general idea of constructing theoretical models that are also executable, pursued over fifty years ago by Newell and Simon.

However, instead of a simulative program, ECB focuses on a more abstract model of the biological system:

1. the processes of abstraction omits from the model those implementation details of the simulative program that have no theoretical value;
 2. the executability of the abstract model permits to expand the class of predictions that can be extracted from the observation of the simulative programs executions.
- ▶ ECB modifies the discovery and corroboration processes of hypotheses on simulated systems in the methodology of simulative AI and AL.

Conclusions

- ▶ ECB resumes the general idea of constructing theoretical models that are also executable, pursued over fifty years ago by Newell and Simon.

However, instead of a simulative program, ECB focuses on a more abstract model of the biological system:

1. the processes of abstraction omits from the model those implementation details of the simulative program that have no theoretical value;
 2. the executability of the abstract model permits to expand the class of predictions that can be extracted from the observation of the simulative programs executions.
- ▶ ECB modifies the discovery and corroboration processes of hypotheses on simulated systems in the methodology of simulative AI and AL.

Conclusions

- ▶ ECB resumes the general idea of constructing theoretical models that are also executable, pursued over fifty years ago by Newell and Simon.

However, instead of a simulative program, ECB focuses on a more abstract model of the biological system:

1. the processes of abstraction omits from the model those implementation details of the simulative program that have no theoretical value;
 2. the executability of the abstract model permits to expand the class of predictions that can be extracted from the observation of the simulative programs executions.
- ▶ ECB modifies the discovery and corroboration processes of hypotheses on simulated systems in the methodology of simulative AI and AL.

Conclusions

- ▶ ECB resumes the general idea of constructing theoretical models that are also executable, pursued over fifty years ago by Newell and Simon.

However, instead of a simulative program, ECB focuses on a more abstract model of the biological system:

1. the processes of abstraction omits from the model those implementation details of the simulative program that have no theoretical value;
 2. the executability of the abstract model permits to expand the class of predictions that can be extracted from the observation of the simulative programs executions.
- ▶ ECB modifies the discovery and corroboration processes of hypotheses on simulated systems in the methodology of simulative AI and AL.

The Epistemology of Computer Simulation (EOCS) (Winsberg 2015):

- ▶ Simulation and Experiment
- ▶ Computer simulations and the structure of scientific theories
- ▶ Fiction
- ▶ Verification and Validation
- ▶ Novel features of EOCS

References

Angius, N. (2014). The problem of justification of empirical hypotheses in software testing. *Philosophy & Technology*, 27(3), 423-439.

Baier C. and Katoen J. P. (2008). *Principles of model checking*. Cambridge, MA: MIT Press.

Cooper R. P. and Guest O. (2014). Implementations are not specifications. Specification, replication and experimentation in computational cognitive modeling. *Cognitive Systems Research*, 27, 4249.

Fisher J. and Henzinger T. A. (2007). Executable cell biology. *Nature Biotechnology*, 25, 1239-1249.

Fisher J., Harel D., and Henzinger, T. A. (2011). Biology as reactivity. *Communications of the ACM*, 54, 72-82.

Johnson-Laird, P. N. (1981). Mental models in cognitive science. In Norman, D. A. (ed.), *Perspectives on Cognitive Science*, pages 147-191. Ablex, Norwood, NJ.

Magnani, L., Nersessian, N., and Thagard, P. (1999). *Model based reasoning in scientific discovery*. Dordrecht: Kluwer.

Newell A. and Simon H.A. (1960). Computer simulation of human thinking. *Science* 134, 2011-17.

Newell A., and Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.

References

- Norton, S., and Suppe F. (2001). Why atmospheric modeling is good science, in C. Miller and P. Edwards (eds.) *Changing the Atmosphere: Expert Knowledge and Environmental Governance*, pp. 88-133. Cambridge, MA: MIT Press.
- Parker, w. (2008b), Computer Simulation through an Error-Statistical Lens, *Synthese*, 163(3), 37184.
- Pylyshyn, Z. W. (1984). *Computation and Cognition: Towards a Foundation for Cognitive Science*. MIT Press, Cambridge, MA.
- Symons, J., and Horner, J. (2014). Software intensive science. *Philosophy & Technology*, 27(3), 461-477.
- Thagard, P. (1984). Computer programs as psychological theories. In Neumaier, O. (ed.), *Mind, Language and Society*, pages 7784. Conceptus-Studien, Vienna.
- Winsberg, E. (1999). Sanctioning Models: The Epistemology of Simulation, *Science in Context*, 12(3), 27592.
- Winsberg, E. (2015). Computer Simulations in Science, *The Stanford Encyclopedia of Philosophy* (Summer 2015 Edition), Edward N. Zalta (ed.).