Application of the Automata Theory to Modelling of Socio-Economic Systems

Diploma Thesis

Barbora Zimmerová

Thesis advisor:
Ing. Marcel Ševela, Ph.D.
Department of Economics

Brno, January 2009
Czech Republic
Acknowledgement

First and foremost, I would like to thank my advisor, Marcel Ševela, for the freedom I had in finding the right direction of this thesis, and his support during this time. I thank him for all the valuable comments, and his incredible patience with me :-). From the long list of my friends and family, special thanks goes to Tomi for his idea to start this studies together, and to František for simply everything beyond.
Declaration

I declare that this thesis is my original work, which I have worked out on my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in the bibliography.

Brno, January 1, 2009
Abstract


This thesis follows quite recent discovery that even very complex phenomena common in nature can emerge from interaction of a large number of agents following very simple behavioural rules, reflected in the automata-based fashion. Besides a literature survey and discussion of an appropriate research methodology for examining models defined in such a way, the thesis specifies, examines and interprets a selected economic system (fundamentalist/chartist stock market) in terms of simple interacting automata, and uses this experimental study to practically evaluate the strengths and weaknesses of the approach.

*Keywords:* finite automata, agent-based economics, modelling, simulation, verification.

Abstrakt


Tato diplomová práce sleduje poměrně nové zjištění, že dokonce i velmi komplexní jevy pozorovatelné okolo nás mohou mít podstatu své složitosti v interakci velkého počtu agentů sledujících velmi jednoduchá pravidla chování, definovatelná na základech teorie automatů. Mimo přehled literatury a diskuzi vhodné metodologie pro zkoumání modelů definovaných tímto způsobem, se práce věnuje specifikaci, analýze a interpretaci zvoleného ekonomického systému (akciového trhu) nahlíženého jako systém interagujících automatů, a na základě této experimentální studie prakticky hodnotí silné a slabé stránky celého přístupu.

*Klíčová slova:* konečné automaty, multiagentní ekonomické modely, modelování, simulace, verifikace.
# Contents

1 Introduction .................................................. 1  
1.1 The problem statement ....................................... 1  
1.2 Aim of the thesis ............................................ 2  
1.3 Thesis outline ............................................... 3  

2 State of the Art .............................................. 5  
2.1 Representative formalisms .................................... 5  
  2.1.1 Finite automata .......................................... 5  
  2.1.2 Cellular automata ....................................... 7  
2.2 Overview of existing work in the field ....................... 10  
  2.2.1 Agent-based computational economics .................... 10  
  2.2.2 Agent-based modelling with cellular automata .......... 12  
  2.2.3 Agent-based modelling with finite automata ............ 15  
2.3 Goals of the thesis revisited ................................ 15  

3 Methodology .................................................. 17  
3.1 Modelling language ........................................... 17  
  3.1.1 Finite automata .......................................... 17  
  3.1.2 Input/Output automata .................................. 18  
3.2 Creation and validation of the model ......................... 22  
3.3 Reasoning techniques ....................................... 22  
  3.3.1 Simulation ............................................... 23  
  3.3.2 Formal verification ...................................... 24  
  3.3.3 Combination of the techniques ........................... 25  
3.4 Tools ....................................................... 26  
  3.4.1 The SPIN model checker .................................. 26  

v
4 Experimental Study

4.1 A fundamentalist/chartist stock market ............................................ 31

4.2 Building of the initial model ......................................................... 33
  4.2.1 Model of a simple trader ....................................................... 34
  4.2.2 Model of a stock ................................................................. 35
  4.2.3 Model of a stock market with simple traders ............................. 35
  4.2.4 Limited amount of speculative money ..................................... 42
  4.2.5 Model of a fundamentalist .................................................... 45
  4.2.6 Model of a chartist ............................................................. 46
  4.2.7 Model of a stock market with both types of traders ................. 47

4.3 Initial analysis of the model .......................................................... 51
  4.3.1 Model .............................................................................. 51
  4.3.2 Simulation ........................................................................ 52
  4.3.3 Hypotheses ....................................................................... 52
  4.3.4 Verification ...................................................................... 55

4.4 Detailed examination of the model and model extensions .......... 57
  4.4.1 Strategies evaluation ............................................................. 57
  4.4.2 Influence of the negotiation time ........................................... 62
  4.4.3 Influence of the initial amount of speculative money ............... 67
  4.4.4 Influence of the stock value changes .................................... 70
  4.4.5 Effect of inter-fundamentalists variability ............................... 72
  4.4.6 Effect of inter-chartist variability ......................................... 79
  4.4.7 Different proportions of the two types of traders .................... 88

5 Discussion of the Results ................................................................. 95
  5.1 Results and observations ............................................................ 95
  5.2 Possible extensions of the model ............................................... 101
  5.3 Strengths and weaknesses of the approach ................................. 101
    5.3.1 S&W of the agent-based paradigm ..................................... 102
    5.3.2 S&W of the automata theory in agent-based modelling ........ 102

6 Conclusion ...................................................................................... 105
  6.1 Possible future directions ........................................................... 106

Bibliography ..................................................................................... 107
A Additional Data 111
A.1 Automata ........................................ 111
A.2 Charts ............................................. 116
   A.2.1 Limited amount of speculative money ........ 116
   A.2.2 Model of a stock market with both types of traders .... 117
   A.2.3 Initial analysis of the model .................. 118
   A.2.4 Strategies evaluation ........................... 120
   A.2.5 Influence of the negotiation time ............. 122
   A.2.6 Influence of the initial amount of speculative money ...... 123
   A.2.7 Effect of inter-fundamentalist variability ........... 124
   A.2.8 Effect of inter-chartist variability ............. 126
   A.2.9 Different proportions of the two types of traders ....... 128

B ProMeLa Models 129
   B.1 Basic model with simple traders ..................... 129
   B.2 Traders with a limited amount of speculative money ..... 130
   B.3 Model of a stock market with two types of traders ..... 131
   B.4 Rating of trader strategies added ................. 135
   B.5 Negotiation time added ............................ 139
   B.6 Stock value changes added ......................... 143
   B.7 Inter-fundamentalist variability added ............. 147
   B.8 Inter-chartist variability added .................. 151
   B.9 The final model (with step count) .................. 156

C Content of the CD 163
Chapter 1

Introduction

“It is not uncommon in the history of science that new ways of thinking are what finally allow longstanding issues to be addressed.”

Stephen Wolfram in A New Kind of Science [18]

1.1 The problem statement

Three centuries ago, science was transformed by the dramatic new idea that rules based on mathematical equations could be used to describe the natural world [18]. Some three decades ago, another such revolution was initiated by the observation that even much more simple types of rules embodied in interacting agents can give rise to incredibly complex and almost random-looking behavioural structures, not different from the phenomena we see in nature, and which we wrongly suppose to emerge from correspondingly complex initial rules [1, 17]. A number of authors over the world are coming to the same conclusion, that the complexity of phenomena around us is not grounded in the complexity of underlying behavioural rules, but is emerging from the interaction of a large number of agents governed with very simple and regular rules, naturally expressible in terms of simple automata.

While in physics, biology and computer science, agent-based reasoning with automata representation helped to disclose the true nature of a number of common phenomena [18], the economics is still standing on the borderline of the possibilities revealed by this discovery. A possible explanation lies, in my opinion, in two main observations I witnessed during the work on this thesis. The first one is the
conservativeness of the traditional economics, which is open to the shift from the united to agent-based representation of economic systems, but still a little sceptical to a more venturous shift from analytical solution of equation-based models to examination of automata-based models capturing the real virtue of agent operation more naturally [1, 18]. The second observation, which most likely seriously reinforces the first one, is the absence of works that would show the way of representing real socio-economic systems in terms of simple interacting automata, and demonstrate the capability of existing tools employed already in other domains of science to effectively examine such models.

1.2 Aim of the thesis

This thesis aims to become one of such works, showing the way of employing the automata theory to modelling and examining non-trivial socio-economic systems, and presenting a compact example of such application. Despite the relative simplicity of the economic system that was selected as a demonstrative example, we feel that we were successful in this goal. There are three main contributions of this thesis to the current state of the art in agent-based reasoning about economic systems with automata-based representation.

First, the thesis includes a compact survey of existing results in the field, and based on the findings from available literature, it designs an appropriate research methodology for building and examining automata-based models of economic systems.

Second, and more importantly, it chooses an exemplary economic system (a financial stock market), identifies its agents (fundamentalistic and chartistic traders), defines their rules, builds their automata-based models, and examines the properties of the system via simulation and formal verification of the composite model and its modifications.

Third and finally, the thesis critically evaluates the findings encountered during the experimental study and relates them to the nature of automata-based formalism employed for their discovery.
1.3 Thesis outline

The text of the thesis is structured into six chapters and three appendixes. The content of the chapters is outlined below.

**Chapter 2 State of the Art** introduces representative automata-based formalisms addressed by this work, and surveys the existing results on their employment in agent-based modelling of socio-economic systems.

**Chapter 3 Methodology** highlights the finite-automata formalism, and its input/output-automata extension, and derives an appropriate research methodology governing the construction and formal analysis of models described by this means.

**Chapter 4 Experimental Study** guides the reader throughout specification, modelling, validation, simulation and formal verification of a selected economic model of a fundamentalist/chartist stock market.

**Chapter 5 Discussion of the Results** summarizes the obtained results and discusses possible upgrades of the model. It critically evaluates the abilities and potential of the automata theory in economics and identifies its weaknesses.

**Chapter 6 Conclusion** closes the thesis with the summary of its contributions and practical impacts, and outlines possible future directions.

**Appendix A Additional Data** contains additional automata and simulation charts supplementing results presented in the main part of the thesis.

**Appendix B ProMeLa Models** consists of the full ProMeLa models of the analysed stock market, following the evolution of the model and its modifications.

**Appendix C Content of the CD** outlines the structure of the data contained on the attached CD.
CHAPTER 1. INTRODUCTION
Chapter 2

State of the Art

“The core of my argument is a claim that informational technologies and processes have critical properties which are imperfectly captured by models in the mechanistic analytical tradition but which are well defined when building blocks of the economy are represented as abstract automata. Real economies contain agents who have adapted to these information technologies, while economists on the whole have not.”

Peter S. Albin in Barriers and Bounds to Rationality—Essays on Economic Complexity and Dynamics in Interactive Systems [1]

2.1 Representative formalisms

There are two main formalisms representing the concepts, both in a variety of different modifications. This section briefly introduces the reader to both of them. The formalism employed in this thesis, namely the finite automata, will be formally defined in Section 3.1.1.

2.1.1 Finite automata

A finite automaton (often called finite state machine, or finite state automaton) [7, 13, 3] is a behavioural model represented as a graph with a finite number of nodes, called states, and oriented edges (arrows) labelled with symbols from a given alphabet. One state is designated the start (initial) state, and some states may be designated as final states.
An example of a finite automaton modelling a simple lifecycle of a lamp is in Figure 2.1. The automaton consists of three states, namely Off, On and KO, and four action symbols. The initial state, Off in this case, is filled with gray colour. There is one final state, KO, depicted with double borderline. This notation is used in the same fashion along the whole text. The model can be interpreted as follows. Whenever a lamp starts its lifecycle, is newly deployed, it is in the Off state. When someone switches it on, it becomes On. In the On state, it can react to two stimuli: switching off, which takes it back to the Off state, or burning out of the bulb, which takes it to the KO state. In the KO state, the lamp can finish its lifecycle (KO is a final state). However, it does not need to finish in this state. If the bulb is replaced with a new one, the lamp gets to the initial state, the Off state again (supposed that switching off is a part of the bulb replacement). More interesting examples follow in the remaining sections.

![Finite automaton modelling a lamp](image)

Figure 2.1: Finite automaton modelling a lamp

Finite automata are an important way to describe certain simple, but highly useful languages called regular languages [7]. Regular languages are the most restricted, and the simplest languages in the Chomsky Hierarchy. Languages in this sense are represented by potentially infinite sets of strings. A finite automaton corresponds to a language in a way that the language consists of exactly those sequences of action symbols (strings) that follow along any transition path from the initial state to a final state (called also run of the automaton). Then we say that the automaton accepts the language. The language accepted by our lamp automaton can be sketched as an infinite set \{switchOn.bulbBurnedOut, switchOn.switchOff.switchOn.bulbBurnedOut, switchOn.bulbBurnedOut.bulbRepaired.switchOn.bulbBurnedOut, ...\}.

Thanks to the simplicity of the formalism, and its ability to produce tractable
Chapter 2. State of the Art

models of even very large systems, finite automata are employed for modelling of the large systems that need to be handled automatically with formal methods, and hence need to be represented in a formal way. Typical applications include modelling of hardware systems, logical circuits, and quite recently also simple software applications.

2.1.2 Cellular automata

Another conceptually simple, but very rich model of nonlinear systems, are the cellular automata (or CA in short). A cellular automaton consist of a (potentially infinite) number of cells arranged typically on a line segment, a circle, a segment of the two-dimensional plane, or a torus, which implies the notion of cell neighbourhood (e.g. the left and right neighbour in case of the line segment). In each period of a simulation, each cell is characterized by a certain state, chosen from a finite set of possibilities (typically represented with different colours).

A cellular automaton (its cells) advances through time, synchronized to a common periodic clock external to the system. The state of each cell in the next period depends on the current state of the cells in a neighbourhood containing it [1].

John Convey’s Game of Life. The best known example of CA is the Game of Life model devised by John Horton Conway in 1970 [20]. The cells of the example are arranged along a two-dimensional plane, each cell in one of two possible states, live or dead. Every cell interacts with its eight neighbours, which are the cells that are directly horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- Any live cell with fewer than two live neighbours dies, as if by underpopulation.
- Any live cell with more than three live neighbours dies, as if by overcrowding.
- Any dead (empty) cell with exactly three live neighbours is populated with a living cell.
- The remaining cells continue unchanged to the next generation.

Despite the simplicity and regularity of the rules, the system evolution exhibits a high level of irregularity and unpredictability. Three snapshots of possible system evolution (dependent on the initial states of the cells) are in Figure 2.2. The first after 200 steps, then after 500 steps and finally after 1000 steps. To reflect at least partly the evolution of the system, the pictures show cells that were black (live) on
Chapter 2. State of the Art

preceding steps in progressively lighter shades of gray.

![Fig 2.2: Three snapshots of the Game of Life [18]](image)

**Figure 2.2: Three snapshots of the Game of Life [18]**

**One-dimensional cellular automata.** Very nice examples of the concept can be presented already on the simplest kind of CA, one-dimensional CA, with cells arranged along a line segment with only two neighbours (left and right) and self. An example of such an automaton is in Figure 2.3, where each row represents a step in time (read from the top to the bottom). Even if the rules for this automaton are very simple (a cell becomes black whenever exactly one of its neighbours is black, otherwise nothing happens), the evolution produces an interesting nested pattern.

![Fig 2.3: Evolution of a simple one-dimensional CA [18]](image)

**Figure 2.3: Evolution of a simple one-dimensional CA [18]**

The power of one-dimensional CA was not much recognized before they started to be extensively studied by Wolfram. In [18], Wolfram provided a number of interesting applications of this simple formalism. He has shown for instance, that one-dimensional CA (with two neighbours and self) with 16 colours for each cell
can compute a task so complicated as a list of successive prime numbers. The computation is depicted in Figure 2.4. The prime numbers are represented with white stripes. In any point in time (row of the evolution), when read from left to right, dark cells determine non-primes and white cells the primes. The sub-figure in the top-left corner stops at the row with the primes 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 and 31.

Figure 2.4: A CA computing a list of successive prime numbers [18]

The ability of CA to compute such a complicated task is not a coincidence. Wolfram showed that even simple one-dimensional CA may exhibit universality. This basically means that any algorithm, any program one can run on a computer or other computational device, whatever complexity it has, can be exactly mimicked with a one-dimensional cellular automaton—a fairly astonishing result.

On the other hand, CA are not primarily intended as a computational formalism. The application of CA is better suited for uncovering of new phenomena and the
Chapter 2. State of the Art

rules that cause them. So it uses to be applied when studying how things evolve, grow, separate and change. In physics and biology, the most typical examples are the growth of crystals, the breaking of materials, fluid flow, fundamental issues in biology (like structure of cells, organs or plants), growth of plants and animals, or biological pigmentation patterns.

2.2 Overview of existing work in the field

This section surveys selected research directions that in different aspects closely relate to the aim of this thesis. It starts with the concept of agent-based computational economics, and follows with the overview of existing work on the realization of this concept with discussed automata-based formalisms.

2.2.1 Agent-based computational economics

The core building block of the agent-based approach is the observation that system complexity does not need to arise from the complexity of the agents, these can be very simple, but from the complexity of their mutual interaction. Sterman in [15] literally says that “the most complex behaviors usually arise from the interactions (feedback) among the components of the system, not from the complexity of the components themselves.”

According to some authors, the agent-based approach finally takes us back to realistic tackling of economic issues in economics of adaptive behaviour and ongoing market processes while circumventing the technical obstacles which forced the forerunners to adopt the static method [10]. The authors of [10] believe that agent-based computational methods provide the only way in which the self-regulatory capabilities of complex dynamic models can be explored so as to advance our understanding of the adaptive dynamics of actual economies. This is essential when trying to understand the major economic disasters, and trying to avoid them, or at least understand the causal effects that led to them.

Bounded rationality. One of the crucial implications of agent-based models, due to the simplicity of agents in particular, is the bounded rationality of agents in the models. Interestingly, this is not understood as an obstacle of the approach, quite the contrary. Already in 1957, Herbert Simon [14] emphasized that individuals are
limited in their knowledge about their environment and in their computing abilities, and moreover that they face search costs to obtain sophisticated information in order to pursue optimal decision rules. Simon argued that, because of these limitations, *bounded rationality* with agents using simple but reasonable or satisficing rules of thumb for their decisions under uncertainty, is a more accurate and more realistic description of human behaviour that perfect rationality with fully optimal decision rules. In the seventies this view was supported by evidence from psychology laboratory experiments of Kahneman and Tversky, showing that in simple decision problems under uncertainty humans do not behave rational in the sense of maximizing expected utility, but their behaviour can be described by *simple heuristics* which may lead to significant biases [6].

**Methodology.** The methodology of agent-based computational analysis follows in a number of steps. Initially, the modeller needs to identify the agents in the system. Examples of possible agents may include individuals (e.g. consumers, workers), social groupings (like families, firmas, government agencies), institutions (like markets, regulatory systems), biological entities (like crops, livestock, forests), or physical entities (like infrastructure, weather, and geographical regions). After identifying agents of modelled economy, the agents are assigned individual cost/utility functions, as well as other characteristics, and the system itself is assigned definitions of the production, pricing, and trade processes driven by interactions among the agents actually residing within the model. Finally, the modeller specifies the initial states of the agents comprising the economy, and formulates the conditions characterizing the state of the economy that is of their interest. This is typically represented by the stable state of the economy (equilibrium state). The modeller can then employ analytical methods that resolve the system and report the values of system parameters in the state of interest.

**Typical applications in economics.** Typical application of the agent-based computational approach to modelling and analysis of socio-economic systems discussed by various authors include [15, 17]:

- Corporate growth and stagnation,
- Business cycles,
- Speculative bubbles,
- The use and reliability of forecasts,
- The design of supply chains in business and other organizations,
Chapter 2. State of the Art

- Project management and product development,
- Civil violence (citizens, rebelling citizens, cops),
- The size of wars (power-law distribution),
- Audience ovation (collective behaviour),
- Residential segregation.

The research in agent-based economics has already uncovered a large number of interesting applications, and presented a number of valuable examples identifying potential rules of the agents. In computational economics, however, agents are modelled with systems of linear and nonlinear equations, and economic problems formulated and resolved with analytical methods. The potential of the automata theory in this domain hence remains unexplored.

2.2.2 Agent-based modelling with cellular automata

The most frequently cited work connecting the concepts of agent-based models with automata-based formalisms is the book of Albin [1], which at the same time makes the first moves in abstracting interacting agents as simple automata, as cellular automata in particular. The choice of cellular automata is in this case governed also with a scientific technique to system analysis, formulated by Wolfram in [18].

Wolfram’s new kind of science. The essential idea of Wolfram’s direction to system analysis is to start with an experiment, with agents assigned very simple and sometimes even random behavioural rules, and then learn from the results. This goes in the opposite direction than the traditional science, which starts from the result, a hypothesis it formulates, and tries to derive the prerequisite conditions that lead to such a behaviour. In Wolfram’s opinion, the traditional science basically tries to find evidence for what it believes is true, and hence is not able to discover the knowledge that was not formulated before, i.e. to uncover new phenomena. Such a new knowledge can come only by coincidence.

Albin’s barriers and bounds to rationality. In the book [1], Albin is mainly concerned with bounded rationality of humans when acting as economic entities, and theoretical limits of reflecting their complexity with automata-based formalisms. He studies the evolution of systems with boundedly-rational agents, and the levels of chaos that the systems exhibit.
The work contains a number of examples, but most of them unfortunately on a very high level of abstraction. One of the examples that are closer to the practical application is a cellular-automata model of microeconomic interaction. The example studies evolution of a system consisting of 100 firms (cells) arranged along a line segment, and reacting to the immediate past of their neighbours to decide on their tendency to capital accumulation (represented with a value they set).

The results of this study are depicted in Figure 2.5, which consists of selected simulations demonstrating different trends. All the simulations are in the format of 100 columns, representing the firms, and 80 rows, representing time steps. The simulations start from different initial distribution of black cells, selected according to a common probability distribution. Black cells are interpreted as increases of the value, white as no increase. Figures a-d show the evolution for homogenous firms and the neighbourhood set of two cells on the left and two cells on the right. The rules are raise $k$ if in the preceding period: (a) 3 firms in the neighbourhood raised $k$, (b) 1, 2, or 3 firms raised $k$, (c) 1 firm raised $k$, (d) 2 or 4 firms raised $k$. And for the remaining figures, if (e) 2 or 3 firms raised $k$, (f) 1 or 3 firms raised $k$. Figures (g) and (h) give simulated data for non-homogenous case, where the firms are divided into 20 columns (industries), the neighbourhood defined as the economic proximity (within the industry), and the rules of raising $k$ whenever: (g) 1 or 2 firms raised $k$, (h) 2 or 4 firms raised $k$. For more detailed description of the example see [1], page 114.

Typical applications in economics. Applications of CA to modelling of socio-economic systems unfortunately do not include many examples. Besides the evolution of a given value that agents raise/decrease based on their environment (demonstrated above), the existing examples use to be concerned with socio-environmental issues. A typical example is urbanic planning studying evolution of inhabiting selected areas based on the properties of the area.

Cellular automata have already shown to be very interesting in general settings [18]. In the domain of economics, however, the research is still being kept on a very general level, concerning itself with more philosophical questions than practical examples. Cellular automata can be very useful in uncovering new phenomena. But to confirm the phenomena, CA are not the best formalism to employ. It is because the only method available to scientists, who want to analyse the models, is the simulation, i.e. observing the evolution of the model. When another computational
Figure 2.5: Application of CA in economics [1]
methods are needed to confirm the observed phenomena, finite automata can come into the play.

### 2.2.3 Agent-based modelling with finite automata

Finite automata are very well suited for agent-based systems modelling. When extended with interaction constructs (input/output actions), each agent can be modelled as a finite automaton and the complete system generated automatically as a composition of these. Since each automaton is basically a graph, and the composite system can be as well, one can employ existing graph techniques to explore the graph and report the information learnt from it.

One of the best known techniques for reasoning about systems modelled in this way is called **formal verification**. In the context of analysing systems formalized as finite automata (or any other equivalent-expressive finite-state formalism), formal verification is the act of proving or disproving hypotheses defining typically the correctness attributes of the system. Currently, there is a number of methods realizing this principle [13, 3], some fully automatic (e.g. model checking), others semi-automatic (e.g. theorem proving).

As distinct to the simulation in case of CA, which can only give us some intuition on the properties of the model, formal verification is able to prove the properties for certain. The two techniques hence complement each other, and can offer a great benefit when combined together.

Unfortunately, in the current literature, there are basically no larger examples employing finite automata to modelling of economic systems. We are aware of only one group that concerns itself with agent modelling with finite automata in the application domain close to economics, the work of DeLoach et. al. [8]. The authors discuss automata-based modelling of agent conversations following different negotiation protocols, like contract or auction protocols. Such examples, however, employ only a small number of agents (one agent of each type), and hence do not give rise to a significant interaction complexity that usually arises when a large number of (potentially similar) agents is included.

### 2.3 Goals of the thesis revisited

In the light of the work overviewed in this chapter, we can identify a gap in existing practical applications of the automata theory to the modelling and analysis of
Chapter 2. State of the Art

economic systems. In this chapter, we have surveyed existing work on two representative formalisms, cellular and finite automata. Since for the finite automata, we have found a larger number of existing analytical techniques, we decided to choose finite automata as the main formalism employed in this thesis.

After defining a research methodology of applying finite automata to system modelling and analysis in the next chapter, we concern ourselves with a definition, model creation, simulation and verification of a selected economic model in terms of finite automata, and discuss the lessons learnt from this study.
Chapter 3

Methodology

"Over time, increasingly sophisticated tools are permitting economic modellers to incorporate procurement processes in increasingly compelling ways. Some of these tools involve advances in logical deduction and some involve advances in computational power."

Leigh Tesfatsion in Agent-Based Computational Economics: A Constructive Approach to Economic Theory [16]

This chapter describes the technical background that allows us to take advantage of the growing computational capacity of today’s computers, and emerging automated formal methods based on this advantage. The chapter formally defines finite automata and its specific variation, Input/Output automata, and derives an appropriate research methodology for examining socio-economic models based on them.

3.1 Modelling language

This section defines the automata-based formalism employed in this thesis. It starts with a formal definition of finite automata introduced informally already in Section 2.1.1.

3.1.1 Finite automata

A finite automaton consists of five constituents, whose naming may slightly differ from author to author. Namely, these are a finite set of states, typically denoted \( Q \),
Chapter 3. Methodology

an alphabet of action symbols, typically Σ, one state marked as the start/initial state, typically q₀, zero or more final/accepting states, typically set F, and a transition function, typically δ. This function takes a state and action symbol as arguments, and returns a state (the target of the transition). Formally, the definition can be written as follows.

**Definition 1 (Finite automaton).** A finite automaton is a 5-tuple $A = (Q, Σ, δ, q₀, F)$ where

- $Q$ is a finite set of states,
- $Σ$ is an alphabet (set of action symbols),
- $δ : Q × Σ → Q$ is a transition function,
- $q₀ ∈ Q$ is an initial state and
- $F$ is a set of final states.

One more important term is a run of an automaton $A = (Q, Σ, δ, q₀, F)$, which is a (potentially infinite) alternating sequence of states and action symbols starting in the initial state $q₀, a₀, q₁, a₁, q₂, \ldots$ where $qᵢ ∈ Q$, and $∀i : δ(qᵢ, lᵢ) = qᵢ₊₁$.

Currently, there is a number of extensions of finite automata, which are better suited for modelling of various systems depending on the application context. In this thesis, we employ an extension distinguishing three types of actions, namely input, output and internal actions, and allowing for non-determinism of the transition set. We refer to this extension as Input/Output automata.

### 3.1.2 Input/Output automata

The Input/Output automata defined in this section (inspired by [2, 4, 11]) modify the basic finite automata model in two aspects. First, actions have an additional semantics differentiating them to input, output, and internal actions. When the automata are understood as interacting agents, each input action represents an attempt to receive a message from another agent, an output an attempt to send a message to another agent, and an internal action a behaviour that is not dependent on other agents in the system. This comes in useful when we compose the automata into a larger model. Then any input/output of an automaton is blocked from occurring until its complement (output/input) becomes active in a different automaton in the system (receiver waiting for the sender / sender waiting for the receiver).

The second aspect that distinguishes Input/Output automata from the standard
finite automata model is the possibility of non-deterministic behaviour. In particular, the transition space is no more defined as transition function \( \delta : Q \times \Sigma \rightarrow Q \), but is given as a transition set \( \delta \subseteq Q \times \Sigma \times Q \). This means that one state can have more then one outgoing transition labeled with the same action symbol, which was not possible before. This is only a minor extension giving us more freedom when modelling real system. This extension has no effect in terms of the expressive power of the formalism.

The formal definition of Input/Output automata follows. Two examples of Input/Output automata are given in Figure 3.1 on the left. In the graphical form (see the example), all input actions are preceded by symbol \( ? \) and all outputs by symbol \( ! \), which helps to distinguish them. Internal actions have no prefix.

**Definition 2 (I/O automaton).** A Input/Output automaton (or I/O automaton) is a 7-tuple \( \mathcal{A} = (Q, \Sigma_{\text{inp}}, \Sigma_{\text{out}}, \Sigma_{\text{int}}, \delta, q_0, F) \) where

- \( Q \) is a finite set of states,
- \( \Sigma_{\text{inp}}, \Sigma_{\text{out}}, \Sigma_{\text{int}} \) are pairwise disjoint sets of input, output and internal actions, respectively; let \( \Sigma = \Sigma_{\text{inp}} \cup \Sigma_{\text{out}} \cup \Sigma_{\text{int}} \) denote the complete alphabet,
- \( \delta \subseteq Q \times \Sigma \times Q \) is a set of labelled transitions,
- \( q_0 \in Q \) is an initial state and
- \( F \) is a set of final states.

As discussed already above, the purpose of distinguishing input, output and internal action is the implied capability of the formalism to model interaction among automata. The actual interaction then emerges through composition of the automata, which can be defined as follows.

Given two I/O automata, we define a product automaton (a finite automaton) representing their joint behaviour and interaction in between. The set of states of the product automaton is defined as a Cartesian product of their state sets, alphabet as a union of the alphabets, initial state as a tuple consisting of their initial states, and the set of final states as a Cartesian product of the two final-state sets. Finally, the transition set consists of all transitions among product states of the automata such that each transition reflects that either (1) one of the automata follows its original transition over an internal action and the other waits, or (2) both automata synchronise on complementary actions (input for one, and output for the other) and form a new joint transition. This is formalized by the following definition, and demonstrated in Figure 3.1 where the right-most automaton represents the
Figure 3.1: Examples of I/O automata and their composition
composition of the two automata on the left. In the example, as well as on other places in the text, product states \((i, j)\) are written in a compact form as \(ij\).

**Definition 3 (Composition).** Let \(A_i = (Q_i, \Sigma_{i,inp}, \Sigma_{i,out}, \Sigma_{i,int}, \delta_i, q^0_i, F_i), i = 1, 2,\) be I/O automata. Then \((Q_1 \times Q_2, \Sigma, \delta, (q^0_1, q^0_2), F_1 \times F_2)\) is a product of \(A_1\) and \(A_2\) iff

- \(\Sigma = (\Sigma_{1,inp} \cup \Sigma_{2,inp}) \cup (\Sigma_{1,out} \cup \Sigma_{2,out}) \cup (\Sigma_{1,int} \cup \Sigma_{2,int})\) and
- \(((q_1, q_2), a, (q'_1, q'_2)) \in \delta\) if
  - \(a \in \Sigma_{1,int} \land (q_1, a, q'_1) \in \delta_1 \land q_2 = q'_2, \) or
  - \(a \in \Sigma_{2,int} \land q_1 = q'_1 \land (q_2, a, q'_2) \in \delta_2, \) or
  - \(a \in (\Sigma_{1,inp} \cap \Sigma_{2,out}) \cup (\Sigma_{1,out} \cap \Sigma_{2,inp}) \land (q_1, a, q'_1) \in \delta_1 \land (q_2, a, q'_2) \in \delta_2.\)

Analogically, we could define the composition of any set of automata (not necessarily two of them). The only tricky part of the definition is the composite transition set. The transition set would now consist of all transitions among product states of the automata such that each transition reflects that either (1) one of the automata follows its original transition over an internal action and all others wait, or (2) exactly two automata synchronise on complementary actions (input for one of them, output for the other) and form a new joint transition, while all others wait.

An example of a composition of three automata at once can be found in Figure 4.5, a composition of four automata in Figure 4.6, both composing automata defined in Figure 4.4. The formal definition of this general composition follows.

**Notation 1.** Let \(\mathcal{I} = \{1, 2, \ldots, n\}\) denote a nonempty finite set of integer indexes. If \(Q_i, i \in \mathcal{I},\) are sets then \(\Pi_{i \in \mathcal{I}} Q_i\) denotes the Cartesian product of the sets \(Q_i\) ordered by the indexes \(i \in \mathcal{I}\). Moreover, for any \(j \in \mathcal{I}, \pi_j\) denotes the projection function \(\pi_j : \Pi_{i \in \mathcal{I}} Q_i \rightarrow Q_j\) returning the \(j\)-th constituent of the \(n\)-tuple \(\pi_j((q_i)_{i \in \mathcal{I}}) = q_j.\)

**Definition 4 (Composition II).** Let \(A_i = (Q_i, \Sigma_i, \Sigma_{i,inp}, \Sigma_{i,out}, \Sigma_{i,int}, \delta_i, q^0_i, F_i)_{i \in \mathcal{I}}\) be a nonempty finite set of I/O automata. Then \((\Pi_{i \in \mathcal{I}}Q_i, \Sigma, \delta, (q^0_1, q^0_2, \ldots, q^0_n), \Pi_{i \in \mathcal{I}}F_i)\) is a product of \(A_i\) in \(\mathcal{I}\) iff

- \(\Sigma = (\bigcup_{i \in \mathcal{I}} \Sigma_{i,inp}) \cup (\bigcup_{i \in \mathcal{I}} \Sigma_{i,out}) \cup (\bigcup_{i \in \mathcal{I}} \Sigma_{i,int})\) and
- \(\delta = \delta_{int} \cup \delta_{sync}\) where
  - \(\delta_{int} = \{(q, a, q') \mid q, q' \in \Pi_{i \in \mathcal{I}} Q_i, \exists j \in \mathcal{I} : a \in \Sigma_{j,int} \land [(\pi_j(q), a, \pi_j(q')) \in \delta_j \land \forall i \in (\mathcal{I} \setminus \{j\}) : \pi_i(q) = \pi_i(q')]\},\)
  - \(\delta_{sync} = \{(q, a, q') \mid q, q' \in \Pi_{i \in \mathcal{I}} Q_i, \exists j_1, j_2 \in \mathcal{I}, j_1 \neq j_2 : a \in (\Sigma_{j_1,int} \cap \Sigma_{j_2,out}) \cup (\Sigma_{j_1,out} \cap \Sigma_{j_2,int}) \land [(\pi_{j_1}(q), a, \pi_{j_1}(q')) \in \delta_{j_1} \land (\pi_{j_2}(q), a, \pi_{j_2}(q')) \in \delta_{j_2} \land \forall i \in (\mathcal{I} \setminus \{j_1, j_2\}) : \pi_i(q) = \pi_i(q')]\}\).
3.2 Creation and validation of the model

The creation of a model in terms of finite automata usually proceeds in an iterative way. First, the modeller needs to decide on an appropriate level of abstraction, applied to the modelled system. The abstraction filters the complexity of the system via highlighting only such aspects that represent the essential behaviour of the system, which is to be studied.

When deciding on the abstraction level, we need to bear in mind that higher abstraction (less details) implies loss of some behavioural information, and lower abstraction (more details) implies higher complexity and size of the final model, which may become unanalyzable due to the size.

After producing an initial model, the model needs to be validated to make sure that the model is correct and corresponds to the intended real system. The validation should test that, first, the model *models the right thing* via checking the appropriateness of the chosen level of abstraction, and second, that it *models the thing right* via checking the syntactic and semantic correctness of the model.

If the model does not behave as expected, it follows to the next iteration, where it is refined (if the abstraction is too coarse-grained) or corrected (if the model is faulty), and validated again. The process finishes as soon as the model passes the validation phase.

Moreover, in case of a complex system, the initial model can be also constructed in an iterative way, starting with a trivial model and iteratively adding complexity to the model, with a validation phase following each refinement step.

**Validation** can be realized by various means. When validating finite-automata models, we can use the same techniques as those discussed already in Section 2.2 as means for analysing automata-based systems, i.e. simulation and formal verification. Simulation can be employed to get the first impression on the model behaviour, and compare it to the expected one, and formal verification to fully examine the model and check its correctness attributes. Both techniques are discussed in more detail in the next section.

3.3 Reasoning techniques

As discussed earlier, there are two classes of reasoning techniques that can be successfully employed to examine automata-based models. The first class consists of
simulation techniques, the second of formal verification techniques. This section discusses these techniques in more detail.

3.3.1 Simulation

Simulation techniques are typically employed in engineering of new technologies to gain insight into the operation of these systems, or to observe their behaviour. They allow for better understanding of causal interdependencies in system behaviour, and hence are essential in creation of mental models of real complex systems.

The central idea of simulation techniques is the *watch and learn* principle. Since watching the real system is very slow, expensive, and sometimes even impossible, simulation techniques use to watch abstract models of the reality, which allows them to deduce new information about system behaviour faster, cheaper, and in larger extent.

In finite-automata models, each observed behaviour of the modelled system is represented with a run of the automaton (alternating sequence of states and transitions starting in the initial state). Since there is often more than one transition outgoing from each state, there can be an enormous number of runs possible in each automaton. Under such conditions, automatic techniques to generate simulation runs are of great value.

There are three main types of simulations connected with finite-automata models, classified according to their autonomousness, namely the random, interactive and guided simulation.

**Random simulation** is a random experimental run of an abstract model. The randomness is in the resolution of choice situations, done whenever there are more alternatives of the future behaviour of the model. A random simulation can be usually assigned an initial seed (typically a number) which resolves the choices in a deterministic (repeatable) way.

**Interactive simulation** relies on a user, who manually resolves the choices observed during the simulation. Interactive simulation stops whenever a choice with more than one option is encountered, and waits until the user select the alternative. This kind of simulation is useful especially when the user wants to get insight in the
alternatives of the system behaviour possible in each state. Note that this information is not visible directly from the model if the model is defined as a composition of many automata.

**Guided simulation** has a different purpose than the two previous. It is employed when a user wants to be guided through the model, following a specific run. Such a run can be reported by some analytical technique (as erroneous for instance). The guided simulation hence allows the user to examine the circumstances under which the specific run occurred, and see how likely it is to occur.

### 3.3.2 Formal verification

Formal verification techniques aim for rigorous proving of the validity of user-specified assumptions on system behaviour, in terms of presence of a good behaviour or absence of a bad behaviour. There exist many techniques implementing this concept [13, 3], some fully automatic (e.g. model checking), others semi-automatic (e.g. theorem proving).

**Model checking.** The best known formal-verification technique for analysis of systems modelled as finite automata (or any equivalent-expressive finite-state formalism) is the *Model Checking* technique [3, 13]. Model checking is an automated state-space exploration technique that, given a finite-state model of a system and a temporal-logic property, systematically checks whether the property holds for (a given initial state in) that model. The technique has two inputs.

- A (finite-state) model of the analysed system. In our case given as a finite-state automaton defined as a composition of a set of previously specified I/O automata.
- A temporal-logic property that formalizes the hypothesis that is to be checked on the model. The hypothesis is understood as the statement that *the temporal property is valid on any possible run of the model (transition path starting in the initial state).*

**Linear Temporal Logic.** The temporal-logic property is given as a formula in a selected temporal logic. One of the most common logics employed for this purpose is the *Linear Temporal Logic (LTL)*, which is going to be used also in this thesis.
Chapter 3. Methodology

LTL is a logic with operators referring to future points in time. In LTL, one can express properties about future-time events, such as that a condition will eventually become true, will stay true forever, or will be true until some special event occurs.

To this end, LTL is built up from a set of atomic prepositions (typically constraints on variables employed in the model, e.g. \((X = 0), (X > 10)\)), the standard boolean operators and and not \((\land, \lor, \neg)\), and two temporal operators next-step and until \((\mathcal{X}, \mathcal{U})\) with the meaning given below.

- The next-step \(\mathcal{X}\) is a unary operator, where \(\mathcal{X}\Phi\) is valid on a run in a model iff the formula \(\Phi\) holds in the next state of the run.
- The until \(\mathcal{U}\) is a binary operator, where \(\Phi \mathcal{U} \Psi\) is valid on a run in a model iff \(\Psi\) becomes true somewhen in the future and until then, \(\Phi\) constantly holds.

Besides these basic operators, one can define a number of useful derived operators. Namely the boolean connectives or and implies \((\lor, \Rightarrow)\), and temporal operators globally and future \((\mathcal{G}, \mathcal{F})\), where \(\mathcal{G}\) states that a formula is globally true in all future states, and \(\mathcal{F}\) states that a formula becomes true in some future state. These derived operators can be defined in terms of the basic ones as the following syntactic shortcuts:

- \(\Phi \lor \Psi \equiv \neg (\neg \Phi \land \neg \Psi)\),
- \(\Phi \Rightarrow \Psi \equiv \neg (\Phi \land \neg \Psi)\),
- \(\mathcal{F}\Phi \equiv \text{true } \mathcal{U} \Phi\),
- \(\mathcal{G}\Phi \equiv \neg \mathcal{F}\neg \Phi\).

Formal definition of LTL, and a detailed explanation of the model-checking technique for verification of LTL formulas can be found in [3, 13].

3.3.3 Combination of the techniques

In the thesis, we combine both simulation and formal verification in the way that brings us the highest value. After building a model and checking its validity, we run a number of simulations that give us the first intuition on the properties of the model. From the simulations, we derive a number of hypotheses, which are then formalized in terms of linear temporal logics, and formally verified using the model-checking technique.
Chapter 3. Methodology

3.4 Tools

One of the main benefits of automata-based formalisms is that they have clearly defined semantics in terms of graph structures, and can be hence easily understood by formal tools that can analyse them fully automatically.

Today, one can choose from a large number of available tools that support both simulation and formal verification of temporal properties defined in various temporal logics. A structured comparison of more than 60 tools can be found at [22].

For the purpose of this thesis, we have selected the SPIN tool, which is currently the most popular tool in the context of LTL model checking of finite-automata models. Besides LTL model checking (including a number of optimisation techniques), the tool has support for both random and interactive simulations. A more detailed description of the tool follows.

3.4.1 The SPIN model checker

Spin is a popular open-source software tool designed for the formal verification of distributed software systems. Its development started at Bell Labs in 1980, and the tool still continues to evolve. In 2001 the tool was awarded the prestigious ACM System Software Award [21, 5].

The functionality. Spin can be used as a full LTL model checking tool, supporting all correctness requirements expressible in linear temporal logic, and also as an efficient on-the-fly verifier for basic safety and liveness properties. Correctness properties can be specified as system or process invariants (using assertions), as linear temporal logic requirements (LTL), or more broadly as general properties in the syntax of never claims.

The tool supports random, interactive and guided simulation, and both exhaustive and partial proof techniques, based on either depth-first or breadth-first search. The tool is specifically designed to scale well, and to handle even very large problem sizes efficiently.

Besides verification and simulation, the tool can check also syntactic validity and logical consistency of models. It reports on deadlocks, unspecified receptions, flags incompleteness, race conditions, and unwarranted assumptions about the relative speeds of processes.
Spin can be used in three main modes:

- as a simulator, allowing for rapid prototyping with a random, guided, or interactive simulations,
- as an exhaustive verifier, capable of rigorously proving the validity of user specified correctness requirements (using partial order reduction theory to optimise the search), and
- as a proof approximation system that can validate even very large system models with fair coverage of the state space.

The tool supports a high level language for specification of system models, called ProMeLa (Process Meta Language). The name of the tool can be then understood as Simple Promela Interpreter (SPIN). The remainder of this section outlines the essential aspects of the ProMeLa language. Full understanding of the language is not necessary for understanding the content of this thesis, as the models are going to be presented also in a graphical form, in terms of I/O automata. Nevertheless, in case of a deeper interest in the topic, we encourage the reader to find a detailed description of ProMeLa in [21, 5].

**ProMeLa language.** A ProMeLa model consists of type declarations, channel declarations, variable declarations and process declaration, sometimes with a special initial process (which is launched as the first one). Every ProMeLa model corresponds to a finite transition model, which means that, first, all constructs are at the end translated into states and transitions, and second, all data referred in the model have finite domains (otherwise we would finish with an infinite-state model). An example of a ProMeLa model is depicted in Figure 3.2, and a number of other examples can be found in Appendix B.

The most important part of a ProMeLa model are the process-type declarations, marked with the proctype keyword. Each process type defines a template of an automaton, and can be also visually drawn in that way. A process type (proctype) consist of a name (in our example T and S), a list of formal parameters (the empty brackets in both T and S), local variable declarations (e.g. \texttt{byte subj; in proctype S}) and a body consisting of a sequence of statements separated with ; and -> symbols.

Process instances can be created in two ways. Either using the run statement at any point in the execution (within any process), or by adding the active keyword in front of the proctype declaration. In the second case, the number of process
mtype { buy, bought, sell, sold };  
chan market = [0] of { mtype, byte };  
short StockPrice = 150;  /* Stock price and its initialisation */

active[2] proctype T () /* Traders */
{
   do
   :: market!buy(_pid); /* Attempt for buying */
      atomic /* Buying processed */
      {
         market?bought(eval(_pid)) ->
         StockPrice = StockPrice + 2;
      }
   :: market!sell(_pid); /* Attempt for selling */
      atomic /* Selling processed */
      {
         market?sold(eval(_pid)) ->
         StockPrice = StockPrice - 2;
      }
   od
}

active[2] proctype S () /* Stocks */
{
   byte subj;
   do
   :: atomic /* Being bought */
      {
         market?buy(subj);
         market!bought(subj);
      };
   atomic /* Being sold */
      {
         market?sell(eval(subj));
         market!sold(subj);
      };
   od
}

Figure 3.2: Example of a simple ProMeLa model
instances is given in square brackets. In our example, we can see that the model consists (is defined as a composition) of four automata, two automata of the type $T$ and two of the type $S$.

The final (automata-based) model is understood as the composition of all process instances (automata) specified in the ProMeLa model. The composition is defined in terms of interleaving semantics with non-deterministic choice, with the exception of communication events that are executed simultaneously (by both communicating parties). ProMeLa supports different means of communication, including handshake, buffered message passing, and communication through shared memory.

In our models, however, we employ only the simplest type of communication, the handshake communication\(^1\), which is realized with exactly the same principle as the communication of I/O automata defined in Section 3.1.2. Then each automaton defined by proctype declaration in a ProMeLa model can be interpreted as an I/O automaton and the full model as their composition with the semantics given above.

\(^1\)In Figure 3.2, the handshake communication is realized via a zero-size channel $market$ to which processes can send messages ($market!buy(pid)$; in proctype $T$) and from which they can receive messages ($market?buy(subj)$; in proctype $S$).
Chapter 4

Experimental Study

“Analysis of very simple and unrealistic models can reveal new theoretical ideas that have broad applicability, beyond the stylized models that produced them. Pressure to make models more realistic (and agents more cognitively sophisticated) is misguided if models become so complex that they are as difficult to interpret as natural phenomena.”

Michael W. Macy and Robert Willer in From Factors to Actors—Computational Sociology and Agent-Based Modeling [12]

This chapter identifies, iteratively models, validates, simulates and verifies an exemplary economic model, of a fundamentalist/chartist stock market. The choice of this example was motivated by a number of existing studies in the domain of agent-based economics, which have identified simple but realistic behavioural rules that can give rise to a nice and complex economic interplay [6, 19]. As distinct to the existing work studying this economic model, we incorporate a modelling approach and computational tools based on the automata theory, which is novel in this domain.

4.1 A fundamentalist/chartist stock market.

In many economic-market models, two important types of traders use to be distinguished, fundamentalists and chartists. Fundamentalists base their expectations about future asset prices and their trading strategies upon market fundamentals and economic factors, such as dividends, earnings, macroeconomic growth, unemployment rates, etc. They tend to invest in assets that are undervalued, that is, whose
prices are below a benchmark fundamental value, and sell assets that are overvalued, that is, whose prices are above the market fundamental value. In contrast, chartists or technical analysts do not take market fundamentals into account but instead base their expectations about future asset prices and their trading strategies upon observed historical patterns in past prices. Technical analysts try to extrapolate observed price patterns, such as trends, and exploit these patterns in their investment decisions. [6]

Abstraction of the market essentials. Following the example of Zeeman [19], we consider a financial stock market consisting of the two types of traders, fundamentalists and chartists, and one type of traded stocks. We assume that fundamentalists know the true value of the stock and buy (sell) when the stock price is below (above) that fundamental value. Chartists are trend followers, buying when price rises and selling when price falls. Each trader holds a given amount of speculative money, available for investments, which can also limit its trading decisions. We suppose a fixed amount of stocks emitted for trading. Thanks to this, the amount of stocks available on the market can be regularly reflected in the stock price. The price raises whenever the amount of available stocks becomes smaller (following a buying operation) and falls whenever the amount becomes larger (following a selling operation).

Trading rules in detail. Each trader implements two operations, the purchase and the sale of a stock. A fundamentalist initiates

- buying if \((\text{StockPrice} < \text{StockValue}) \&\& (\text{Money} > \text{StockPrice})\), and
- selling if \((\text{StockPrice} > \text{StockValue})\).

A chartist initiates

- buying if \((\text{PriceTrend} == \text{‘raise’}) \&\& (\text{Money} > \text{StockPrice})\), and
- selling if \((\text{PriceTrend} == \text{‘fall’})\).

The computation of the price trend is based on the price evolution history of the last five price changes. If at least four of the last five changes are in the positive direction (price raises), the trend is set to \text{‘raise’}. If at least four of the last five changes are in the negative direction (price falls), the trend is set to \text{‘fall’}.

Additionally, at the end of each buying/selling operation, we need to update
• the stock price, i.e. increase it in case of buying, decrease in case of selling, and
• the price trend, i.e. update the price evolution history and compute the new price trend.

Model analysis. It was observed already by other authors [19, 6] that fundamentalists act as market stabilizers, while chartists as market destabilizers. When both types are present in an equal merit, the market price tends to converge to the value given by fundamentalists. When more chartists are present, the market price tends to diverge in either direction, and the divergence is further forced by the chartists, which may lead to a complete destabilization of the market.

Besides confirming this findings, we are going to be interested in what type of trading strategy (fundamentalist/chartist) brings higher gain and under what conditions, i.e. if it is influenced by the proportions of traders, variability inside the groups, alternative trading strategies, etc. We will study the stock price evolution under different conditions. We are interested not only in the convergence/divergence process, but also in the position and shape of the evolution curve, including its amplitude for instance.

We follow the methodology defined in the previous chapter, first constructing the initial model and then introducing its refinements bringing it closer to reality. Each refinement step is accompanied with a number of simulations, formulation of hypotheses and their formal verification. All the verifications were performed with the help of the SPIN tool introduced above. The verification reports are all part of this thesis on an attached CD, together with all the models discussed in this chapter (see Appendix C).

4.2 Building of the initial model

To guide even an unexperienced reader throughout the process of initial-model construction, we follow an iterative process with very simple refinement steps. Our model is first going to consists of only two types of automata, one modelling the behaviour of a simple (general) trader, the second reflecting the lifecycle of a stock item, which can be bought or sold by the traders. This model is then refined with the specific behaviour that distinguishes the traders to fundamentalists and chartists.
4.2.1 Model of a simple trader

Suppose a simple trader interested in buying and selling of stocks, both mediated by a market shared with the remaining traders. After initiation, the trader decides which of the tasks (buy/sell) to perform first and after completing it, it returns to the initial state and continues with deciding on the next task. Both buying and selling task is finished with the update of the global stock price. See Figure 4.1 depicting the model of such a trader, and Appendix B.1 containing the ProMeLa model of the trader as a process named $T$.

Notice the similarity of the model in Figure 4.1 to the I/O automaton in Figure 3.1. There are a few minor differences in notation. First, in Figure 4.1, action symbols are preceded with a name of the mediating channel, namely market. However, the market channel is of zero size (see its definition in the ProMeLa model, Appendix B.1), which means that it realizes the handshake communication assumed implicitly when defining I/O automata. Thanks to this, market?action / market!action here has the same meaning as input/output actions ?action / !action used and defined in Section 3.1.2.

Second, input and output actions can be assigned input/output parameters. For a variable var and constant cons, both !action(var) and !action(cons) represent an output event passing the value of the variable, resp. constant; ?action(var) is an input event assigning the received value to the variable; and ?action(cons) is an input event that is executable only if the passed value equals to the specified constant. All these constructs can be realized in I/O automata as well, and can be hence understood as syntactic shortcuts.

Third, internal actions (i.e. $\text{StockPrice} = \text{StockPrice} + 2$ in Figure 4.1) can be now written as expressions with both local and global effects. For the syntax of possible expressions see ProMeLa specification in [21, 5]. The expressions could be again encoded into simple I/O automata model.

In summary, the buying and selling tasks of a trader in Figure 4.1 are defined as follows. The buying starts with sending a buy request buy assigned with the identification of the trader (process identifier .pid), and waiting for receiving the buy response bought addressed to the trader. Execution of this sequence is possible only if there is an automaton (modelling stock) that receives the buy request and confirms it with sending the bought response with the right id. After that, the trader updates global stock price and returns to the initial state. The selling task
can be read analogically.

Notice that there are no conditions guarding the buying/selling decisions, like realization of a selected strategy or limitation by the amount of available money. The trader can buy a stock whenever a stock is available (ready to synchronize with the trader), and can sell a stock whenever it currently owns one (there is a stock that can confirm the selling with the id of the trader). This logics will be better visible after specification of the stock model, and the composition of these, discussed in Section 4.2.3.

![Figure 4.1: Model of a simple trader](image)

### 4.2.2 Model of a stock

The model of a stock is in Figure 4.2. There are two important states in the model, the initial state symbolizing that the stock is free to be bought, and the state 2 reflecting that the stock is bought and ready to be sold. The important point is that along the first step of the loop (characterizing the lifecycle of the stock), the stock remembers the identification of its buyer, in variable subj, and in the remainder of the loop checks that it communicates only with its owner.

Note that this represents a parametric description of the model. If there is only one trader that can be interested in the stock, the model of the stock defines the first automaton in Figure 4.3. If there are two traders, then the model defines the second automaton in Figure 4.3. And so on, and so forth.

### 4.2.3 Model of a stock market with simple traders

The model of the complete market is defined as a composition of the automata of traders and stocks with the handshake semantics, formally defined in Section 3.1.2.
Chapter 4. Experimental Study

Figure 4.2: Model of a stock

Figure 4.3: An instance of the stock model for one/two/three traders
Chapter 4. Experimental Study

Table 4.1: Influence of the atomicity on the size of composite models

<table>
<thead>
<tr>
<th>Components</th>
<th>ATOMICITY</th>
<th>NO ATOMICITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>States</td>
<td>Transitions</td>
</tr>
<tr>
<td>2 Tr. x 1 Stock</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>2 Tr. x 2 Stocks</td>
<td>53</td>
<td>67</td>
</tr>
<tr>
<td>3 Tr. x 3 Stocks</td>
<td>216</td>
<td>1.297</td>
</tr>
<tr>
<td>4 Tr. x 4 Stocks</td>
<td>4,096</td>
<td>40,961</td>
</tr>
<tr>
<td>5 Tr. x 5 Stocks</td>
<td>100,000</td>
<td>1,500,000</td>
</tr>
</tbody>
</table>

The handshake composition forces the components of the system to synchronize on complementary actions (input and output with the same name) and allows arbitrary interleaving of all remaining (internal) actions.

Since the arbitrary interleaving of independent actions can enormously increase the size of the composite model, ProMeLa allows the modeller to define some sequences of actions as **atomic**. Every atomic sequence of actions is then (in the composition) performed as one action, i.e. without interleaving with other actions in between, which significantly reduces the size of the final model. The final ProMeLa model of the market, including atomicity, can be found in Appendix B.1. Table 4.1 summarizes the difference in size of the market model, with different numbers of components (traders and stocks), and with/without atomicity.

To get a better picture of the composite-model behaviour, Figures 4.5 and 4.6 depict two composite models of the simple market with two traders. When the number of traders is fixed to two, concrete models of traders and stocks (after evaluation of possible passed values) can be written as in Figure 4.4. In the figure, each **buy-bought** and **sell-sold** sequence is defined as atomic (separated by states with a bold borderline). Then the model in Figure 4.5 demonstrates the composite model of the market consisting of two traders and one stock item, and Figure 4.6 depicts the composite model of a market with two traders and two stocks.

**Non-determinism.** It is worth noting that, as distinct to other types of models, the final automata-based models are highly non-deterministic, allowing at each point in time a number of alternative actions to happen. One can see in the composite model in Figure 4.6, that traders often compete with each other when buying a stock, each having an equal chance to succeed. The model even includes runs, where one
Chapter 4. Experimental Study

Figure 4.4: Models of market parts

Figure 4.5: The composition of two traders with one stock
Figure 4.6: The composition of two traders with two stocks
of the traders is constantly preempted by others. There is no judge in the model, who would guard that the market is equally fair to all traders. Such a judge can be, however, encoded in the model during formal verification.

Simulation. The following charts depict exemplary simulations of the market with different number of agents, namely 2 traders and 2 stocks in Figure 4.7, 2 traders and 10 stocks in Figure 4.8, and 2 traders and 15 stocks in Figure 4.9.

The simulations show that the number of available stocks delimits the borders of stock-price evolution. Based on the models, we can compute the lower bound as \( \text{Init} \) and the upper bound as \( \text{Init} + 2 \times \text{Stocks} \), where \( \text{Init} \) denotes the initial stock price, and \( \text{Stocks} \) the number of emitted stocks (number of stock automata in the model).
Chapter 4. Experimental Study

Model validation. Before bringing the model closer to economic reality, we need to validate that the model is constructed correctly. First, we validated the model syntactically by the SPIN tool, and then semantically via checking the absence of deadlock states (states from which the model cannot move any further), and validity of the finding about the bounds of the stock price (to check that the model behaves as expected).

In the report below, as well as in the whole chapter, **In LTL** is followed with a formalization of the hypothesis in terms of linear temporal logic (LTL), **In ProMeLa** with definitions of included atomic prepositions and transcription of the formula in ProMeLa, and **Result in X** with the verification result on a model X. The results are accompanied with the time needed to report them. For instance, 1h20m6s means 1 hour, 20 minutes and 6 seconds; 2.1s means 2.1 seconds.

<table>
<thead>
<tr>
<th>H-01</th>
<th>The model is deadlock-free.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Result in 2Tx2S:</strong></td>
<td>Valid! Confirmed after 2.1s.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>H-02</th>
<th>The stock price is always above the initial value.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In LTL:</strong></td>
<td>$G(StockPrice &gt; 150)$</td>
</tr>
<tr>
<td><strong>In ProMeLa:</strong></td>
<td>#define spGt150 (StockPrice &gt; 150) $\Box spGt150$</td>
</tr>
<tr>
<td><strong>Result in 2Tx2S:</strong></td>
<td>Not valid! Error trail found in 1.9s.</td>
</tr>
</tbody>
</table>

The error trail shows that the formula is invalid already in the initial state.
We need to ask the stock price to be above or equal to the initial value.

**In LTL:** \( G(StockPrice \geq 150) \)

**In ProMeLa:**
```
#define spGEt150 (StockPrice >= 150)
[]spGEt150
```

**Result in 2Tx2S:** Valid! Confirmed after 2.2s.

---

**H-03** The stock price is always below the maximal value of 154.

**In LTL:** \( G(StockPrice \leq 154) \)

**In ProMeLa:**
```
#define spLEt154 (StockPrice <= 154)
[]spLEt154
```

**Result in 2Tx2S:** Valid! Confirmed after 2.0s.

---

### 4.2.4 Limited amount of speculative money

In reality, each trader operates with a limited amount of speculative money. This can be reflected in a variable \( Money \) local to each trader in the model (Appendix B.2, Figure 4.10). The trader can start a new purchase only if \( Money > StockPrice \). After buying a stock, the price of the stock is subtracted from its \( Money \), and after selling, it is added to its \( Money \).

To keep the system in balance, we suppose that the price which the trader pays/obtains is defined as an average of the price before and after the operation. In particular, if a trader buys a stock, it pays \( StockPrice+1=StockPrice'–1 \), where \( StockPrice \) represent the price before the operation and \( StockPrice' \) is the price after the operation. Now, the reader can see the reason for raising the stock price by two—this allows us to charge the middle price, while staying with natural numbers.

Moreover, in the refined model, we start to observe the number of stocks held by each trader in a global array named \( Stocks[] \).

**Simulation.** The simulations in Figures 4.11 and 4.12 show that as distinct to the previous model, the stock price cannot grow so high anymore, due to the limit on the amount of money available to each trader. Additionally, Figures A.6 and A.7 in appendix depict the evolution of a number of stocks held by the traders, which is independent between the traders, and connected only to the stock-price evolution in a way that the sum of the stocks follow the same trend as the stock price.
Figure 4.10: Model of a trader

Figure 4.11: Example of stock-price changes for the 2Tx10S settings
Model validation. The following list summarizes the results of semantic validation of the refined model. The syntactic validation did not report any errors. Again, the results only confirm the correctness of the model, they do not report any findings about the modelled market. The market will be analysed after completing the initial model, within verification parts of the sections.

**H-01** The model is deadlock-free.

**Result in 2Tx10S:** Not valid! Deadlock found in 1.6s. The error trail shows that the reason lies in an insufficient number of stocks available to the traders. Both traders get blocked after checking the validity of the condition `Money>StockPrice` and cannot move further. We modify the model so that the number of stocks becomes sufficient.

**Result in 2Tx12S:** Valid! Confirmed after 4m41s.

**H-02** The stock price is always above the initial value.

**In LTL:** $\mathcal{G} (StockPrice \geq 150)$

**In ProMeLa:**
```
#define spGEt150 (StockPrice >= 150)
[]spGEt150
```

**Result in 2Tx12S:** Valid! Confirmed after 5m15s.

**H-03** The stock price is always below the maximal value of 174.
In LTL: $G(\text{StockPrice} \leq 174)$

In ProMeLa: #define spLEt174 (StockPrice <= 174)

Result in 2Tx12S: Valid! Confirmed after 5m19s.

H-04 The money amount of a trader cannot fall to zero or below. If it could fall below zero, it would indicate that the trader paid more for a stock than expected when starting the purchase.

In LTL: $G(\text{Money} > 0)$

In ProMeLa: #define moneyGt0 (Money > 0)

Result in 2Tx12S: Not valid! Error trail found in 1.8s.

In LTL: $G(\text{Money} \geq 0)$

In ProMeLa: #define moneyGEt0 (Money >= 0)

Result in 2Tx12S: Valid! Confirmed after 5m18s.

Starting from this model, the available computer memory is no more sufficient to perform the exhaustive verification. We need to employ an approximative technique for verification (we employ the BitState technique implemented in SPIN), which does not guarantee the full coverage of the model. On the other hand, notice that for the $G(\text{StockPrice} \geq 150)$ property, no error trail was found within the state-space of more than 5 million states (over 5 minutes needed), while when $G(\text{StockPrice} > 150)$ was considered, the error trail was found in less than 3 seconds. This gives us a fair evidence that the property is really valid.

All models following after this section were validated in the same fashion. To keep the text concise, from now on, we include only the validations that reported unexpected results.

4.2.5 Model of a fundamentalist

Let us continue in the refinement of the trader model via adding more conditions guarding their decision on buying/selling. This gives rise to separate models of fundamentalists and chartists. Each fundamentalist is assigned an individual constant value (local variable StockValue=200). The buying process is now guarded
by the condition \((\text{StockPrice}<\text{StockValue})\&\&(\text{Money}>\text{StockPrice})\) and the selling process by the condition \((\text{Stocks}[\_\text{pid}]=0)\&\&(\text{StockPrice}>\text{StockValue})\), see Figure 4.13.

![Diagram](image)

Figure 4.13: Model of a fundamentalist

### 4.2.6 Model of a chartist

Chartists decide on buying/selling of stocks according to the price trend observed in the past evolution. In our case, a chartist is interesting in buying a stock if at least four of the last five price changes are in the positive direction (price rises), and it sells if at least four of the last five price changes are in the negative direction (price falls).

This concept is realized with the observation of price trend within the global \texttt{PriceTrend} variable, initially set to a neutral value 1. A chartist starts the buying process whenever \((\text{PriceTrend}=2)\&\&(\text{Money}>\text{StockPrice})\) and selling whenever \((\text{Stocks}[\_\text{pid}]=0)\&\&(\text{PriceTrend}=0)\). The \texttt{PriceTrend} variable is set according
Chapter 4. Experimental Study

to the history record of the last five price changes saved in the PT[] array, which needs to be updated anytime a price changes.

Figures 4.14–4.16 depict the main part of the chartist model, and separately the price-trend update in case of price raise (after buying) and fall (after selling). The full model of a chartist extended with trend update is in Appendix A.1, Figure A.1, and the full model of a fundamentalist in Figure A.2.

![Diagram of model of a chartist with trend update as a single transition](image)

Figure 4.14: Model of a chartist with trend update as a single transition

4.2.7 Model of a stock market with both types of traders

The model of the stock market with both types of traders is in Appendix B.3. In addition to the previous versions of the model, it contains one more process
Figure 4.15: Price trend update when price raises
Figure 4.16: Price trend update when price falls
Chapter 4. Experimental Study

definition. It is the initial process (Figure 4.17) responsible for initialisation of the price-trend history in a way that it implies the neutral value of the price-trend indicator $\text{PriceTrend} = 1$.

Figure 4.17: Model initialisation

Simulation. The following simulation demonstrates a typical evolution of the stock price in the fundamentalist/chartist stock market consisting of 2 chartists, 2 fundamentalists and 50 stocks. The corresponding evolution of the number of stocks held by the traders can be found in Appendix A.2, Figure A.8. Since the convergence of the market to a stable state, where $\text{StockPrice} = 200$, is very fast, the evolution of the number of stocks held by the traders is not very interesting (does not show any clear trend).

Model validation. The validation of the model reported a deadlock state (a state from which the model cannot move any further). The reported deadlock points out that the model can (correctly) stop in the equilibrium state—when neither chartists
Chapter 4. Experimental Study

Figure 4.18: Stock-price evolution for the 2Cx2Fx50S settings

nor fundamentalists are interested in buying or selling of stocks. Since this is not an error in the model, and we do not want it to be reported again, we indicated such a situation in the model with valid end states in individual automata (see Figures A.3–A.5 in appendix, end states have double borderline). The modified model then passes the validation without deadlock reporting.

H-01 The model is deadlock-free.

Result in 2Cx2Fx12S: Not valid! Deadlock found in 2.4s.
The error trail shows that this is because the market can stop in the equilibrium state.
Result in 2Cx2Fx12S modified: Valid! Confirmed after 4m12s.

4.3 Initial analysis of the model

After completing the initial model (see Appendix B.3 for the ProMeLa version), we start to examines the model a bit more thoroughly, and try to derive some findings about the modelled fundamentalist/chartist market.

4.3.1 Model

We consider three versions of the initial model, differing in the number of traders of each type. In all three models, the traders start with the initial amount
Chapter 4. Experimental Study

of speculative money set to $\text{Money}=1000$, and all the fundamentalists confirming to $\text{StockValue}=200$. The market starts with the initial stock price set to $\text{StockPrice}=150$. Besides, the models have the following settings:

1. **Model 6Cx6Fx50S** consists of 6 Chartists, 6 Fundamentalists, and 50 Stocks.
2. **Model 2Cx6Fx50S** consists of 2 Chartists, 6 Fundamentalists, and 50 Stocks.
3. **Model 6Cx2Fx50S** consists of 6 Chartists, 2 Fundamentalists, and 50 Stocks.

All the three models have been validated before starting the analysis.

4.3.2 Simulation

We have performed a large number of simulations, showing common trends. Typical examples of the simulations are presented in this section.

All three models (6Cx6Fx50S, 2Cx6Fx50S and 6Cx2Fx50S) typically shortly converge to the stable state, where $\text{StockPrice}=\text{StockValue}$. The stock price can shortly oscillate around the stock value before getting stabilized, which is caused by the involved chartists (see Figure 4.19). One can see in Figures 4.21 and 4.23 that higher proportion of chartists causes higher and longer oscillation period before price stabilization. It would be interesting to check how high can the oscillations be (can the stock price get higher than 230 or lower than 180 for instance?), and if the price always converges to the stock value, or can diverge in some cases. We formulate such hypotheses and verify them formally later in this section.

Charts depicting the evolution of the total number of stocks held by fundamentalists and chartists (Figures 4.20, 4.22 and 4.24) show how the chartists (with a small delay) follow the fundamentalists in their buying/selling behaviour. Detailed charts for the number of stocks held by individual fundamentalists and chartists can be found in Appendix A.2, Figures A.9–A.12. One can notice that the traders in our simulations are in total never interested in more than 30 stock items, even if there are 50 of them available on the market. We will check if this is true globally.

4.3.3 Hypotheses

The simulations uncovered a number of possible trends. We formulate the observations in a number of hypotheses, which are formally verified below with the SPIN tool. Each hypothesis has a short identification name, which is used to refer to the hypothesis along the thesis.
Chapter 4. Experimental Study

Figure 4.19: Stock-price evolution for the 6Cx6Fx50S settings

Figure 4.20: Number of stocks held by the traders for the 6Cx6Fx50S settings

Figure 4.21: Stock-price evolution for the 2Cx6Fx50S settings
Chapter 4. Experimental Study

Figure 4.22: Average number of stocks held by the traders for the 2Cx6Fx50S settings

Figure 4.23: Stock-price evolution for the 6Cx2Fx50S settings

Figure 4.24: Average number of stocks held by the traders for the 6Cx2Fx50S settings
• **H-11** The stock price always converges to the value 200.
• **H-12** The stock price is always lower than 230. Notice that theoretically, it could grow up to 250, since there are 50 stocks available, and enough money at traders’ disposal.
• **H-13** Except the initial phase, the stock price cannot fall below 180.
• **H-14** The traders in total will never be interested in more than 30 stocks.

### 4.3.4 Verification

The following verification reports summarize the results of hypotheses verification. The reports show that if the market consists of equal proportions of traders, or higher portion of fundamentalists, the market price always converges to the fundamental value, under the condition that the fundamentalists are active on the market. If the number of chartists outbalances, the stock price starts to deviate from the fundamental. The same can be visible on the upper bound of the stock price. While on 6Cx6Fx50S, the stock price is always below 230, and on 3Cx9Fx50S even below 210, in case of 9Cx3Fx50S, the price can reach the value 230. Additionally, we checked a limit on the lower bound of the stock price in case of 6Cx6Fx50S, and shown that the traders in the model will never be (in total) interested in more than 35 stocks, which is interesting while they have enough money for much more.

**H-11** The stock price always converges to the value 200.

In **LTL**: \( \mathcal{F} \mathcal{G} (\text{StockPrice} = 200) \)

In **ProMeLa**: #define spEt200 (StockPrice==200)

\( <>[] \text{spEt200} \)

**Result in 6Cx6Fx50S**: Not valid! Error trail found in 2.7s.

The error trail shows that this appears when fundamentalists are passive on the market (are constantly preempted by chartists). We modify the formula so that we check the convergence only on the runs where fundamentalists (at least half of them) are active as well.

In **LTL**: \( (\mathcal{F} \mathcal{G} \text{activeFund}) \Rightarrow (\mathcal{F} \mathcal{G} (\text{StockPrice} = 200)) \)

In **ProMeLa**: #define aF (Stocks[0]>0 && Stocks[1]>0 && Stocks[2]>0)

\( <>[] \text{aF} \Rightarrow (<>[] \text{spEt200}) \)

**Result in 6Cx6Fx50S**: Valid! Confirmed after 10m29s.

In **LTL**: \( (\mathcal{F} \mathcal{G} \text{activeFund}) \Rightarrow (\mathcal{F} \mathcal{G} (\text{StockPrice} = 200)) \)
Chapter 4. Experimental Study

In ProMeLa:  #define aF (Stocks[0]>0)
             (<>[]aF) -> (<>[]spEt200)

Result in 6Cx2Fx50S: Not valid! Error trail found in 5m43s.
The error trail shows that chartists can take the market out of equilibrium by
following the trend set up on their own.

H-12  The stock price is always lower than 230.

In LTL:  G(StockPrice < 230)
In ProMeLa:  #define spLt230 (StockPrice < 230)
              <>[]spLt230
Result in 6Cx6Fx50S: Valid! Confirmed after 10m25s.
Result in 9Cx3Fx50S: Not valid! Error trail found in 3.8s.
Result in 3Cx9Fx50S: Valid! Confirmed after 7m33s.

In LTL:  G(StockPrice < 210)
In ProMeLa:  #define spLt210 (StockPrice < 210)
              <>[]spLt210
Result in 3Cx9Fx50S: Valid! Confirmed after 7m32s.

H-13  Except the initial phase, the stock price cannot fall below 180.

In LTL:  G((StockPrice > 180) ⇒ (G StockPrice > 180))
In ProMeLa:  #define spGt180 (StockPrice > 180)
              [](spGt180 -> []spGt180)
Result in 6Cx6Fx50S: Valid! Confirmed after 10m40s.

In LTL:  G((StockPrice > 190) ⇒ (G StockPrice > 190))
In ProMeLa:  #define spGt190 (StockPrice > 190)
              [](spGt190 -> []spGt190)
Result in 6Cx6Fx50S: Not valid! Error trail found in 8m48s.

H-14  The traders in total will never be interested in more than 30 stocks.

In LTL:  G stocksLEt30
In ProMeLa:  #define stocksLEt30 (Stocks[0]+...+Stocks[11] <= 30)
              []stocksLEt30
Result in 6Cx6Fx50S: Not valid! Error trail found in 2.3s.
In LTL: $\mathcal{G} \text{ stocksLEt35}$

In ProMeLa: #define stocksLEt35 (Stocks[0]+...+Stocks[11] <= 35)

[]stocksLEt35

Result in 6Cx6Fx50S: Valid! Confirmed after 10m24s.

This was confirmed also via deadlock checking for the appropriate number of stocks.

### 4.4 Detailed examination of the model and model extensions

When comparing the results of the previous section to the findings published by other authors, we can confirm the same observations. This means, that thanks to the previous section, we have semantically evaluated our model, and confirmed that it behaves as expected. From now on, we go beyond the results published elsewhere, and try to examine the market model more thoroughly.

#### 4.4.1 Strategies evaluation

Up to now, the model could be examined only via comparison of the number of stocks held by individual traders, and evolution of the global stock price. In addition to these, we now introduce a metric evaluating the profitability of the behaviour of individual traders. Consider the following two metrics:

- **Profit** computed at each point in time as the difference of the current wealth (wealth in owned stocks, counted as $\text{Stocks[trader]}*\text{StockPrice}$, and current amount of speculative money) to the initial wealth (initial amount of speculative money).

- **Rating** computed iteratively after each purchase (or sale) for all traders in a way that we add (resp. subtract) one point per stock to (resp. from) each trader owning some stocks. In case of purchase, the actual buyer does not gain a point for the last stock bought within the purchase, to keep the computation in balance. Otherwise, a trader constantly buying and selling a stock would permanently increase its rating.

Even if it is not obvious at first glance, the simulations below helped us to uncover, that both profit and rating output basically the same curves (see Figures 4.29 and 4.30). It can be shown also mathematically that they reflect the same trend with
only one slight difference. The profit function treats slightly differently the actual buyer (seller) from the remaining stock holders, whereas the rating function treats all traders equally. The assymmetry in the profit function is caused by the simple fact that whenever a trader buys a stock, it pays \( \text{StockPrice} + 1 \) but raises its profit with \( \text{StockPrice} + 2 \) plus the raise per each held stock (where \( \text{StockPrice} \) denotes the price before the operation). Hence the increase of its profit is by one higher then the increase of the profit of other traders with the same amount of stocks.

Moreover, the rating function is better suited for iterative computation because it reuses information from the previous iterations. Therefore from now on, we use the rating function as the main evaluation mechanism. Models of the traders updated with rating computation are in Figures 4.25 and 4.26, and the full ProMeLa model in Appendix B.4.

**Figure 4.25: Model of a fundamentalist with rating**

**Simulation.** The charts in Figures 4.29 and 4.30 depict the evolution of profit and rating in comparison to the stock-price evolution and stocks held by the traders. One can see that the rating (profit) of both fundamentalists and chartists is very similar. At some points, chartists are better off, at other points fundamentalists are evaluated better.
Even more detailed charts can be found in appendix Figures A.13–A.16, depicting the rating (profit) of individual traders. Even if all the traders of each type follow the same strategy, we can observe some variability in their ratings, in case of chartists only in the position of the curves, in case of fundamentalists also in their shape.

**Verification.** We have verified two hypotheses, one regarding the difference in ratings of the two groups, one regarding the difference in rating and number of stocks inside the groups. We confirm that on different runs of the model, the two groups of traders are evaluated differently. No of them has significantly better rating. Regarding the second hypothesis, we show that within both groups, the variability between two traders of the same type can raise to about 50–60 points in rating, which is quite a high number. Regarding stocks, a difference of 4 stocks was discovered among fundamentalists, but was shown impossible among chartists.

While the questions of the second hypothesis ask about the *existence* of a behaviour (run), while model checking proves properties for all runs, we need to formulate the properties in a negative way, and if a property is proved invalid, to interpret it as a confirmation of the existence.
Chapter 4. Experimental Study

Figure 4.27: Stock-price evolution for the 6Cx6Fx50S settings

Figure 4.28: Number of stocks held by the traders for the 6Cx6Fx50S settings

Figure 4.29: Profit of the traders for the 6Cx6Fx50S settings
H-21  From some point in time, the average rating of fundamentalists is and remains higher/lower than the average rating of chartists.

In LTL: $\mathcal{F} \mathcal{G} (\text{avgRatF} > \text{avgRatC})$
In ProMeLa: 
- \text{define avgRatF} ((\text{Rating}[0]+...+\text{Rating}[5])/6)
- \text{define avgRatC} ((\text{Rating}[6]+...+\text{Rating}[11])/6)
- \text{define ratFGtC} (\text{avgRatF} > \text{avgRatC})
- \text{<>[]ratFGtC}

Result in 6Cx6Fx50S: Not valid! Error trail found in 2.4s.
Similarly, an error was found also in 6Cx2Fx50S and 2Cx6Fx50S models.

In LTL: $\mathcal{F} \mathcal{G} (\text{avgRatF} < \text{avgRatC})$
In ProMeLa: 
- \text{define ratFLtC} (\text{avgRatF} < \text{avgRatC})
- \text{<>[]ratFLtC}

Result in 6Cx6Fx50S: Not valid! Error trail found in 2.4s.
Similarly, an error was found also in 6Cx2Fx50S and 2Cx6Fx50S models.

H-22  The inter-fundamentalist variability is higher/lower than the inter-chartist variability. Can it happen that two traders of the same type at one point differ in 4 stocks, or 50 points in rating?

In LTL: $\neg (\mathcal{F} (\text{Rating}[x] > \text{Rating}[y] + 50))$
In ProMeLa: 
- \text{define ratVarF} (\text{Rating}[0] > \text{Rating}[1]+50)
- \text{<>[]ratVarF} — in a negative way

Result in 6Cx6Fx50S: Not valid, i.e. such a trail exists! Found in 15m41s.
Chapter 4. Experimental Study

!(<>ratVarC) — in a negative way

Result in 6Cx6Fx50S: Not valid, i.e. such a trail exists! Found in 15m55s.

In LTL: \(\neg(F(Rating[x] > Rating[y] + 60))\)

In ProMeLa: #define ratVarF (Rating[0] > Rating[1]+60)
!(<>ratVarF) — in a negative way

Result in 6Cx6Fx50S: Valid, i.e. such a trail not found! Confirmed after 16m39s.

!(<>ratVarC) — in a negative way

Result in 6Cx6Fx50S: Valid, i.e. such a trail not found! Confirmed after 16m36s.

In LTL: \(\neg(F(Stocks[x] > Stocks[y] + 4))\)

In ProMeLa: #define stoVarF (Stocks[0] > Stocks[1]+4)
!(<>stoVarF) — in a negative way

Result in 6Cx6Fx50S: Not valid, i.e. such a trail exists! Found in 15m46s.

!(<>stoVarC) — in a negative way

Result in 6Cx6Fx50S: Valid, i.e. such a trail not found! Confirmed after 24m35s.

The main observation of this section is, that there is no significant difference in the profitability of the chartistic and fundamentalistic strategies. The following sections however show that this is true only under quite an unrealistic assumption—that the market very quickly converges to the stable state. We will see that when we introduce some realistic non-regularities, which keep the market for a bit longer outside the stable state, then the ratings of the two groups start to heavily differ.

4.4.2 Influence of the negotiation time

The traders were modelled in a way that each buying/selling operation consists of two separate actions, the first denoting the start (buy action) and the second the finish (bought action) of the operation. This is not a coincidence, while this feature allows us to study the effect of prolonged negotiation time during buying/selling, via allowing a time delay between these two actions. Up to now, the two actions needed to occur consecutively due to the atomic construct in the stock model. Let
us now remove the construct (see the modified model in Appendix B.5) and study the effect of introduced negotiation time.

**Simulation.** We can see in Figure 4.31 that the negotiation time has one main implication. It shifts the effect of buying/selling (stock price changes) further to the future, where it creates irregular peaks in stock-price evolution. Thanks to this, the market is kept for a longer time outside the stable state\(^1\). However, it still reaches the stable state after some time, and stays in it from that point on.

If the peaks are not too concentrated, we can say that the only significant effect of the negotiation time is the prolonged time before the convergence, and hence it gives us more time to observe the behaviour of the two types of traders. We can see in Figures 4.32 and 4.33 that while the number of stocks held by chartists and fundamentalists is very similar, the rating of the two strategies starts to be significantly different, better for fundamentalists as they are the ones that determine the trend followed by chartists.

![Figure 4.31: Stock-price evolution](image)

When the negotiation time is allowed, we should discuss what price the buyers (sellers) pay (obtain) when the price changes during the negotiation. There are two possible strategies, either the price valid when the operation is initiated, or the price valid when the operation is finished. Note that in our case (where the buyers/sellers

\(^1\)We have observed that ten of ten simulations of the model needed in between of 110 and 600 steps to reach the stable state, half of them needed more then 300 steps, whereas in the previous model, ten of ten simulations reached the stable state in less than 50 steps.
Chapter 4. Experimental Study

Figure 4.32: Number of stocks held by the traders

Figure 4.33: Rating of the traders
cannot back away from the operation after starting it) the pricing strategies influence the presented charts only if the amount of money held by the traders during simulation reaches the level that limits their decisions. From Figures A.17 and A.18 in Appendix A.2 (depicting selected traders for better clarity) we can see that in our simulation, this is not the case.

Still, verification in the next section shows that the speculative money of the traders can fall to the amount where it starts to influence traders’ behaviour, so we should make our decision of the pricing system clear. We decide for fixing the price valid when the operation is initiated, because the price after finishing the operation does not any more need to fulfil the condition under which the trader started the operation. The modification of a trader model is outlined in Figure 4.34 for a fundamentalist.

![Figure 4.34: Model of a fundamentalist fixing the price](image)

**Verification.** Despite the modification, the verification shows that the model exhibits very similar properties to the previous model. The only small difference is that the stock price evolution has a bit higher amplitude (can fall below 180 even...
Chapter 4. Experimental Study

after the initial period) and the traders demand a bit higher number of stocks (up to 40, comparing to 35 in the previous model).

**H-11** The stock price always converges to the value 200.

In LTL: \( (\mathcal{F} \mathcal{G} \text{activeFund}) \Rightarrow (\mathcal{F} \mathcal{G} (\text{StockPrice} = 200)) \)

In ProMeLa: #define aF (Stocks[0]>0 && Stocks[1]>0 && Stocks[2]>0)  
\( (<>[\!]aF) \rightarrow (<>[\!]\text{spEt200}) \)

Result in 6Cx6Fx50S: Valid! Confirmed after 9m15s.

**H-12** The stock price is always lower than 230.

In LTL: \( \mathcal{G} (\text{StockPrice} < 230) \)

In ProMeLa: #define spLt230 (StockPrice < 230)  
\( <>[\!]\text{spLt230} \)

Result in 6Cx6Fx50S: Valid! Confirmed after 8m22s.

**H-13** Except the initial phase, the stock price cannot fall below 180.

In LTL: \( \mathcal{G} ((\text{StockPrice} > 180) \Rightarrow (\mathcal{G} \text{StockPrice} > 180)) \)

In ProMeLa: #define spGt180 (StockPrice > 180)  
\( [\!]\text{spGt180} \rightarrow [\!]\text{spGt180} \)

Result in 6Cx6Fx50S: Not valid! Error trail found in 6m15s.

The time needed to find the error trail indicates that even if the property is not valid in general, in most cases it holds. The next verification shows that a slight modification of the limit is already sufficient to make the property valid.

In LTL: \( \mathcal{G} ((\text{StockPrice} > 175) \Rightarrow (\mathcal{G} \text{StockPrice} > 175)) \)

In ProMeLa: #define spGt175 (StockPrice > 175)  
\( [\!]\text{spGt175} \rightarrow [\!]\text{spGt175} \)

Result in 6Cx6Fx50S: Valid! Confirmed after 8m19s.

**H-14** The traders in total will never be interested in more than 35 stocks.

In LTL: \( \mathcal{G} \text{stocksLEt35} \)

In ProMeLa: #define stocksLEt35 (Stocks[0]+...+Stocks[11] <= 35)  
\( [\!]\text{stocksLEt35} \)
Result in 6Cx6Fx50S: Not valid! Error trail found in 6m15s.

In LTL: $\mathcal{G} \ stocksLEt40$

In ProMeLa: #define stocksLEt40 (Stocks[0]+...+Stocks[11] <= 40)

[] stocksLEt40

Result in 6Cx6Fx50S: Valid! Confirmed after 8m24s.

4.4.3 Influence of the initial amount of speculative money

It would be interesting to see, how much the initial amount of speculative money influences the behaviour of the traders. There are two borderlines that actually matter, the lower bound on the initial amount of speculative money that allows the market to function healthily, and the upper bound which is so high that adding more money to the traders have no effect on their behaviour.

Simulation. Let us keep the negotiation time for the simulations, so that the market becomes a little longer in motion. We have performed a number of simulations which showed that the lower bound is somewhere between $M=500$ and $M=550$. When the initial amount of money is below $M=500$, the stock price does not even reach StockPrice==200, and hence the market gets stuck in a situation where all traders have no money left and are not motivated to sell the owned stocks. But when $M$ becomes a little higher, already for $M=550$, the market starts to exhibit the same patterns visible in case of $M=1000$ (see Figures 4.35–4.37) even if the amount of speculative money often reaches the level limiting the decisions of the traders (see Figures A.19 and A.20 in Appendix A.2).

The upper bound on the amount of speculative money that still influences behaviour of the traders can be guessed already from simulations of the previous section. We have observed that when $M=1000$, the money of individual traders hardly reaches the level limiting their decisions. Below, we verify this observation formally.

Verification. The verification shows what is roughly the upper bound on the initial amount of speculative money giving the traders enough freedom to decide according to their essential rules, irrespective of money. We check, on models with different initial money, the condition that the momentary amount of money held by any trader (we consider one representative fundamentalist and one representative chartist) never falls below 200. Surprisingly, this bound is significantly different for
Chapter 4. Experimental Study

Figure 4.35: Stock-price evolution (for $M=550$)

Figure 4.36: Number of stocks held by the traders (for $M=550$)

Figure 4.37: Rating of the traders (for $M=550$)
the two types of traders. While the chartists need about $M=2200$, fundamentalists are content already with $M=1200$.

**H-31** We want to check what initial amount of speculative money is needed so that momentary money held by a trader never falls below 200.

**In LTL:** $\neg (F ((MoneyFund < 200) \lor (MoneyChart < 200)))$

**In ProMeLa:**

```promelag
#define mfLt200 (Money[0] < 200)
#define mcLt200 (Money[6] < 200)
!(<>(mfLt200 || mcLt200))
```

**Result in 6Cx6Fx50S $M=1000$:** Not valid! Error trail found in 2.4s.

**Result in 6Cx6Fx50S $M=2000$:** Not valid! Error trail found in 7m16s.

The time needed for finding the trail indicates that such a situation is not very likely to occur. Compare it to the time needed for finding the trail in case of $M=1000$, and see that when we raise $M$ a little above 2000, it becomes valid.

**Result in 6Cx6Fx50S $M=2200$:** Valid! Confirmed after 9m38s.

**H-32** We want to check what initial amount of speculative money is needed so that momentary money held by a fundamentalist never falls below 200.

**In LTL:** $\neg (F (MoneyFund < 200))$

**In ProMeLa:**

```promelag
#define mfLt200 (Money[0] < 200)
!<>(mfLt200)
```

**Result in 6Cx6Fx50S $M=1000$:** Not valid! Error trail found in 6m37s.

The time needed for finding the trail indicates that such a situation is not very likely to occur. A little more is already sufficient.

**Result in 6Cx6Fx50S $M=1200$:** Valid! Confirmed after 8m49s.

**H-33** We want to check what initial amount of speculative money is needed so that momentary money held by a chartist never falls below 200.

**In LTL:** $\neg (F (MoneyChart < 200))$

**In ProMeLa:**

```promelag
#define mcLt200 (Money[6] < 200)
!<>(mcLt200)
```

**Result in 6Cx6Fx50S $M=1000$:** Not valid! Error trail found in 2.4s.

**Result in 6Cx6Fx50S $M=2000$:** Not valid! Error trail found in 7m16s.

**Result in 6Cx6Fx50S $M=2200$:** Valid! Confirmed after 9m36s.
4.4.4 Influence of the stock value changes

We have observed in the previous sections that irrespective of various modifications of the model, the market price stably converges to the stock value assumed by fundamentalists, which is constant along the time. It would be interesting to see, if the same is true even if the stock value evolves along the time.

We modify the model in a way that every ten steps the stock value (of the fundamentalists) becomes by one higher. We again keep the negotiation time in the model, so that the market gives us more time to observe its behaviour. The modified ProMeLa model can be found in Appendix B.6.

Simulation. We have run a number of simulations to observe the reaction of the market to price value changes. A typical example is presented in Figures 4.38–4.40. Notice two main observations. First, the stock price converges to the stock value as expected. We verify this observation below. Second, the ratings of the two groups, fundamentalists and chartists, again follow the same pattern, positive for fundamentalists.

![Figure 4.38: Stock-price evolution](image)

Verification. We have verified the first observation, that the stock price converges to the stock value even if the stock value is not constant along the time. This was confirmed by the verification.
Figure 4.39: Number of stocks held by the traders

Figure 4.40: Rating of the traders
The stock price converges to the stock value. I.e. there is always a point in time from which on, the stock price is constantly very close to the stock value.

In LTL: \( \mathcal{F} \mathcal{G} (\neg 10 < (SP - SV) \land (SP - SV) < 10) \)

In ProMeLa:

```c
#define spsvNeg10 (-10 < (StockPrice-StockValue))
#define spsvPos10 ((StockPrice-StockValue) < 10)
<>[]((spsvNeg10 && spsvPos10)
```

Result in 6Cx6Fx50S: Valid! Confirmed after 9m48s.

4.4.5 Effect of inter-fundamentalists variability

In this and the following section, we introduce a significant non-regularity to the model. We abandon the assumption of trader homogeneity within the groups, and start to examine the model with heterogeneous fundamentalists and heterogeneous chartists. This section is dedicated to the first one.

One could notice that regarding the rating of the strategies, the fundamentalists are substantially better off. This is natural since they know the real fundamental value of the stocks, and hence have no delay in getting closer to it. Let us take this advantage away from them, and assign each fundamentalist a different value from a predefined distribution (keeping the mean value equal to 200). We examine two distributions of the stock value, one inspired by normal distribution, i.e. in case of six traders as (175, 191, 199, 201, 209, 225), and one reflecting uniform distribution, i.e. for six traders as (175, 185, 195, 205, 215, 225).

To implement this modification, we define the fundamentalist proctype (in the ProMeLa model) in a parametric way. Whenever a new instance of the proctype is created, it is passed the stock value that it should follow. All instances are launched right at the beginning of the model execution (within the init process), so the semantics of the model composition remains unchanged. Figure 4.41(a) depicts the model of the init process which launches the instances of trader models. Chartists are passed only the id used to reference them, fundamentalists are passed also their individual stock value. The init process finishes with setting go=1, which gives the created instances the signal that they can start to execute. Note that since the heterogeneity of traders is itself a substantial distribution for the model, we decide to omit the negotiation time and keep the stock value stable in the model, to make the interpretation of the results easier. The final model can be found in Appendix B.7.
Chapter 4. Experimental Study

(a) Fundamentalists variability
(b) Chartists variability added

Figure 4.41: Initialisation of models with variability
Chapter 4. Experimental Study

Simulation. The most common behaviour (named simulation A) is depicted in Figures 4.42–4.44, and an additional chart with the evolution of stocks held by fundamentalists in Figure A.21 in Appendix A.2. In the simulation, the stock price stops around the value 200\(^2\), where all fundamentalists with the value below 200 have no stocks any more, and all fundamentalists with the value above 200 have no money left to buy more. At the same time, the price trend for all the traders is such that they are not motivated to buy or sell more stocks.

The second example (simulation B, Figures 4.45–4.47) demonstrates that it no more needs to be the case that the stock price converges to 200. The stable situation described for simulation A (fundamentalists below the final price not interested to buy more, fundamentalists above no money to buy more, chartists not motivated to

\(^2\)Note that the oscillation is due to the fact that the stock values are not necessarily even values, while the stock price can be just even.

74
buy/sell) can be valid even if the final stock price is different from 200. In this case, the final stock price is equal to 204, which is in between of the values of the fourth and fifth fundamentalist (values 201 and 209).

Both simulation A and simulation B were performed on the model with normal distribution of stock values. Simulation C (Figures 4.48–4.50) is a variation of simulation B for the uniform distribution of stock values among fundamentalists. One can notice that due to larger gaps between fundamentalist values close to 200, the stock price has higher chance to stabilize more far from the mean value. In this case, the final value is 214, which is again in between of the values of the fourth and fifth fundamentalist (values 205 and 215).
Chapter 4. Experimental Study

Figure 4.46: Rating of the traders (simulation B)

Figure 4.47: Rating of individual fundamentalists (simulation B)

Figure 4.48: Stock-price evolution (simulation C)
Figure 4.49: Rating of the traders (simulation C)

Figure 4.50: Rating of individual fundamentalists (simulation C)
Chapter 4. Experimental Study

**Verification.** Having the modified model, we verify some of the hypotheses verified previously on the homogenous model. Hypotheses H-12 and H-13 finished with the same result, but surprisingly, H-21 did not. In particular, while in Section 4.4.1 it was found that no group of traders seem to have globally better rating, now the verification shows that at the end, fundamentalists always finish with better rating. The reason can be, in our opinion, that in the previous model, the market converged so quickly that it often stopped before the rating of fundamentalists could gain the lead.

In addition to the previously defined hypotheses, we formulate one new hypothesis formalizing that the fundamentalists with the stock value closer to the mean value use to have better rating than the others. This is shown correct for both normal and uniform distribution of stock values.

<table>
<thead>
<tr>
<th>H-12</th>
<th>The stock price is always lower than 230.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In LTL:</strong></td>
<td>$\Box (\text{StockPrice} &lt; 230)$</td>
</tr>
<tr>
<td><strong>In Promela:</strong></td>
<td>#define spLt230 (StockPrice &lt; 230) [\spLt230]</td>
</tr>
<tr>
<td><strong>Result in 6Cx6Fx50S normal:</strong></td>
<td>Valid! Confirmed after 31m44s.</td>
</tr>
<tr>
<td><strong>Result in 6Cx6Fx50S uniform:</strong></td>
<td>Valid! Confirmed after 31m40s.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>H-13</th>
<th>Except the initial phase, the stock price cannot fall below 180.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In LTL:</strong></td>
<td>$\Box (\text{StockPrice} &gt; 180) \Rightarrow (\Box \text{StockPrice} &gt; 180))$</td>
</tr>
<tr>
<td><strong>In Promela:</strong></td>
<td>#define spGt180 (StockPrice &gt; 180) [\spGt180 \Rightarrow \spGt180]</td>
</tr>
<tr>
<td><strong>Result in 6Cx6Fx50S normal:</strong></td>
<td>Valid! Confirmed after 31m25s.</td>
</tr>
<tr>
<td><strong>Result in 6Cx6Fx50S uniform:</strong></td>
<td>Valid! Confirmed after 32m10s.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>H-21</th>
<th>From some point in time, the average rating of fundamentalists is and remains higher/lower than the average rating of chartists.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In LTL:</strong></td>
<td>$\Box G (\text{avgRatF} &gt; \text{avgRatC})$</td>
</tr>
<tr>
<td><strong>In Promela:</strong></td>
<td>#define avgRatC ((Rating[0]+...+Rating[5])/6) #define avgRatF ((Rating[6]+...+Rating[11])/6) #define ratFGtC (avgRatF &gt; avgRatC) [\spGt180 \Rightarrow \spGt180]</td>
</tr>
</tbody>
</table>

78
Chapter 4. Experimental Study

Result in 6Cx6Fx50S normal d.: Valid! Confirmed after 31m8s.
Result in 6Cx6Fx50S uniform d.: Valid! Confirmed after 31m55s.

In LTL: $FG(\text{avgRatF} < \text{avgRatC})$
In ProMeLa: #define ratFLtC (avgRatF < avgRatC)
<> [] ratFLtC

Result in 6Cx6Fx50S normal d.: Not valid! Error trail found in 3.9s.
Result in 6Cx6Fx50S uniform d.: Not valid! Error trail found in 4.0s.

H-51 Fundamentalists with the stock value closer to 200 have better rating, i.e. the fundamentalist with stock value 199 (or 195 in case of uniform distribution) has better rating than the one with stock value 175.

In LTL: $FG(ratF1 > ratF2)$
In ProMeLa: #define ratFGtF (Rating[8] > Rating[6])
<> [] ratFGtF

Result in 6Cx6Fx50S normal d.: Valid! Confirmed after 32m24s.
Result in 6Cx6Fx50S uniform d.: Valid! Confirmed after 31m54s.
For comparison, verification of the opposite direction $FG(ratF1 < ratF2)$ reported an error in 4.1s, for both normal and uniform distribution.

4.4.6 Effect of inter-chartist variability

Up to now, all chartists determined their price-trend following strategy in the same way. If at least four of the last five price changes were in the positive direction (price rise), price trend was set to PriceTrend=2 which motivated them to buy more stocks. If at least four of the last five price changes were in the negative direction (price fall), price trend was set to PriceTrend=0 which motivated them to sell their stocks. Otherwise, price trend was set to a neutral value PriceTrend=1.

Besides this strategy, which we call strategy 1, we define two new strategies in the following way:

- **Strategy 0**: buy if all 5 last changes are positive, sell if all 5 last changes are negative.
- **Strategy 1**: buy if at least 4 of last 5 changes are positive, sell if at least 4 of last 5 changes are negative.
- **Strategy 2**: buy if at least 3 of last 5 changes are positive, sell if at least 3 of last 5 changes are negative.
Figure 4.51 demonstrates this modification in an automata-based fashion, for the case of price raise. The full model in ProMeLa is in Appendix B.8.

**Simulation.** The three strategies can be assigned to chartists in a number of different ways. The most intuitive one, reflected in model called model 001122 in this section, is to assign the first two chartists strategy 0, the next two strategy 1 and the last two strategy 2. An initial process reflecting this distribution is in Figure 4.41(b), and charts of an exemplary simulation are presented below.

The most important chart is in Figure 4.54, depicting average ratings of the chartists following the three strategies, comparing to the average rating of fundamentalists. We have performed a large number of simulations to find out, if there is some stable ordering of the three rating curves.

At first, when looking only at the first 200–300 steps, we did not find any common pattern. See Figures A.24 and A.25 in Appendix A.2, which have different ordering of rating curves comparing to Figure 4.54.

However, when we considered a bit longer period of time, the rating curves started to show a common ordering with the lead of strategy 0. See Figure 4.55 for the long run of the initial simulation, and appendix Figures A.26 and A.27 for the long-run continuations of the alternative simulations.

A next distribution of strategies we tried is 000111 (three chartists following strategy 0 and three chartists following strategy 1). This model, as well as model 000000, showed a strong tendency to converge faster than the model 001122. While in 001122 only 3 of 10 simulations converged in less than 300 steps (it was in 140, 240 and 220 steps), in 000111 all 10 of 10 simulations converged in less than 100 steps (usually around 70), and in 000000 all 10 of 10 simulations converges even in less than 60 steps (usually around 40). Thanks to the fast stabilization of the model, the strategies have not much time to surface, and the ratings are hence very similar. See the charts in Figures 4.56–4.59.

The last two simulations (Figures 4.60–4.63) consider the opposite distribution of strategies, all chartists following strategy 2. We can see that the first simulation exhibits the same pattern as the simulation of model 001122 (observed in 7 of 10 simulations), and the second one the pattern visible in model 000111 (observed in 3 of 10 simulations, stopping after 65, 93, 62 steps). In both types of patterns, the situation stabilizes when the fundamentalists have the assets reflecting their value,
Figure 4.51: Price trend update when price raises
Chapter 4. Experimental Study

Figure 4.52: Stock-price evolution (model 001122)

Figure 4.53: Rating of the traders (model 001122)

Figure 4.54: Rating of the traders with respect to their strategies (model 001122)
Figure 4.55: Rating of the traders w.r.t. strategies in long run (model 001122)

Figure 4.56: Stock-price evolution (model 000111)

Figure 4.57: Rating of the traders (model 000111)
CHAPTER 4. EXPERIMENTAL STUDY

Figure 4.58: Stock-price evolution (model 000000)

Figure 4.59: Rating of the traders (model 000000)
and chartists are either interested in buying and do not have the money, or are interested in selling and have no more stocks to sell.

![Figure 4.60: Rating of the traders (model 222222)](image1)

![Figure 4.61: Stock-price evolution (model 222222)](image2)

**Verification.** In addition to the hypotheses H-12, H-13 and H-21, verified already on previous models, we have added hypotheses H-61 and H-62 studying the rating of chartist strategies. The hypotheses verification shows that carefull following of trends (strategy 0) is in long term more profitable than the impetuous trend following (strategy 2), which can seem to be surprising. The reason could be, in our opinion, in the point that carefull trend following protects the traders during the time when the price oscilates around the equilibrium price, which is the time when the chartists in general loose the most.
Chapter 4. Experimental Study

Figure 4.62: Rating of the traders (model 222222, alternative simu.)

Figure 4.63: Stock-price evolution (model 222222, alternative simu.)
H-12 The stock price is always lower than 230.

In LTL: $\mathcal{G} (\text{StockPrice} < 230)$
In ProMeLa: `#define spLt230 (StockPrice < 230)`
\[
\langle\rangle spLt230
\]
Result in 6Cx6Fx50S 001122: Valid! Confirmed after 1h12m11s.
Result in 6Cx6Fx50S 000000: Valid! Confirmed after 32m22s.
Result in 6Cx6Fx50S 222222: Valid! Confirmed after 1h13m01s.

H-13 Except the initial phase, the stock price cannot fall below 180.

In LTL: $\mathcal{G} ((\text{StockPrice} > 180) \Rightarrow (\mathcal{G} \text{StockPrice} > 180))$
In ProMeLa: `#define spGt180 (StockPrice > 180)`
\[
\langle\rangle (spGt180 \rightarrow \langle\rangle spGt180)
\]
Result in 6Cx6Fx50S 001122: Valid! Confirmed after 1h14m55s.
Result in 6Cx6Fx50S 000000: Valid! Confirmed after 32m35s.
Result in 6Cx6Fx50S 222222: Valid! Confirmed after 21m31s.

H-21 From some point in time, the average rating of fundamentalists is and remains higher/lower than the average rating of chartists.

In LTL: $\mathcal{F} \mathcal{G} (\text{avgRatF} > \text{avgRatC})$
In ProMeLa: `#define avgRatC ((Rating[0]+...+Rating[5])/6)`
`#define avgRatF ((Rating[6]+...+Rating[11])/6)`
`#define ratFGtC (avgRatF > avgRatC)`
\[
\langle\rangle ratFGtC
\]
Result in 6Cx6Fx50S 001122: Valid! Confirmed after 1h10m25s.
Result in 6Cx6Fx50S 000000: Valid! Confirmed after 31m43s.
Result in 6Cx6Fx50S 222222: Valid! Confirmed after 1h10m45s.

In LTL: $\mathcal{F} \mathcal{G} (\text{avgRatF} < \text{avgRatC})$
In ProMeLa: `#define ratFLtC (avgRatF < avgRatC)`
\[
\langle\rangle ratFLtC
\]
Result in 6Cx6Fx50S 001122: Not valid! Error trail found in 6.1s.
Result in 6Cx6Fx50S 000000: Not valid! Error trail found in 4.0s.
Result in 6Cx6Fx50S 222222: Not valid! Error trail found in 4m45s.
The ratings of the chartist strategies at the end always confirm that strategy 0 is better than strategy 2.

In LTL: $\mathcal{F} \mathcal{G} ((\text{ratC}0 > \text{ratC}2))$

In ProMeLa: 
```
#define ratC0GtC2 ((Rating[0]+Rating[1]) > (Rating[4]+Rating[5]))
<>[]ratC0GtC2
```

Result in 6Cx6Fx50S 001122: Not valid! Error trail found in 1h13m45s.

The error trail shows that if the price converges quickly to the final value, then the traders with strategy 2 can take the lead and keep it to the end; notice that the traders with strategy 2 have typically much worse start than those with strategy 0, and they take the lead throughout the time. On the other hand, more than one hour was needed to find an error trail, which means that in most of the cases, the hypothesis is valid.

The ratings of the chartist strategies at the end always confirm that strategy 0 is worse than strategy 2.

In LTL: $\mathcal{F} \mathcal{G} ((\text{ratC}0 < \text{ratC}2))$

In ProMeLa: 
```
#define ratC0LtC2 ((Rating[0]+Rating[1]) < (Rating[4]+Rating[5]))
<>[]ratC0LtC2
```

Result in 6Cx6Fx50S 001122: Not valid! Error trail found in 6.1s.

This is mainly to demonstrate the time needed to find an error trace in this and the previous verification, even if performed on the same model.

### 4.4.7 Different proportions of the two types of traders

The last section of this chapter is dedicated to the examination of different proportions of fundamentalists and chartists in the model, including variability of both types of traders (model in Appendix B.9).

We consider three versions of the model, with the following number of traders and stocks:

1. **Model 12Cx12Fx50S** consists of 12 Chartists, 12 Fund., and 50 Stocks.
2. **Model 6Cx12Fx50S** consists of 6 Chartists, 12 Fund., and 50 Stocks.
3. **Model 12Cx6Fx50S** consists of 12 Chartists, 6 Fund., and 50 Stocks.
The chartists follow strategies 001122 in case of 6 chartists, and 00001112222 in case of 12 chartists. Fundamentalists have values (175, 191, 199, 201, 209, 225) in case of 6 fundamentalists and (170, 184, 192, 196, 198, 200, 201, 204, 208, 216, 230) in case of 12 fundamentalists.

**Simulation.** Simulations of the first model, model 12Cx12Fx50S, show an analogical pattern observed already for the 6Cx6Fx50S model in Figure 4.52. The situation is similar also for model 6Cx12Fx50S and 12Cx6Fx50S with the difference that the stock price evolution in model 12Cx6Fx50S has much higher amplitude than in model 6Cx12Fx50S and 12Cx12Fx50S.

We can again see that when the proportion of chartists becomes significantly higher than the number of fundamentalists, the market price starts to deviate from the fundamental value. The model 12Cx6Fx50S was not sufficient to demonstrate the divergence, so we included one more model with 15 chartists, model 15Cx6Fx50S, which already reaches the limits of the market settings, namely the available number of stocks. See Figures A.28 and A.29 in Appendix A.2, comparing the availability of stocks in the 12Cx12Fx50S and 15Cx6Fx50S models, bearing in mind that the second one includes less traders than the first one.

Besides these expectable results, there is one new interesting point. In all simulations in this section, the stock price is in its evolution most of the time above the line of 200, even if one would expect symmetry around 200 since it is the mean value and trader strategies are defined symmetrically. We are not really sure about the explanation of this observation (confirmed also with formal verification).

![Figure 4.64: Stock-price evolution (model 12Cx12Fx50S)](image-url)
Chapter 4. Experimental Study

Figure 4.65: Average rating of the traders (model 12Cx12Fx50S)

Figure 4.66: Stock-price evolution (model 6Cx12Fx50S)

Figure 4.67: Average rating of the traders (model 6Cx12Fx50S)
Chapter 4. Experimental Study

Figure 4.68: Stock-price evolution (model 12Cx6Fx50S)

Figure 4.69: Average rating of the traders (model 12Cx6Fx50S)

Figure 4.70: Stock-price evolution (model 15Cx6Fx50S)
Chapter 4. Experimental Study

Figure 4.71: Average rating of the traders (model 15Cx6Fx50S)

**Verification.** For the sake of better comparability with the initial analysis, the verifications are executed on the 6Cx6Fx50S, 9Cx3Fx50S and 3Cx9Fx50S models. The first two verifications also on the 12Cx12Fx50S model. The verification results are analogical to the previous ones, with only the slight difference that the setting of 9Cx3Fx50S was no more sufficient to observe the divergence, and that we detected the necessity of a bit higher number of stocks demanded on the market.

This section includes one additional hypothesis, formalizing the observation about the asymmetric positioning of the stock-price curve around the mean value 200. We can see on the 6Cx6Fx50S model in Figure 4.52 that it may take a while before the curve moves up. So in the hypothesis, we check that the curve starts to be within asymmetric bounds after 200 steps. The verification guessing the bounds reveals that while the lower bound is somewhere between 185–190, the upper bound is between 225–230, which confirms the asymmetry.

**H-12** The stock price is always lower than 230.

**In LTL:** $\mathcal{G}(\text{StockPrice} < 230)$

**In ProMeLa:**
```
#define spLt230 (StockPrice < 230)
<>[]spLt230
```

**Result in 6CxF6Fx50S:** Valid! Confirmed already in Section 4.4.6.

**Result in 9CxF3Fx50S:** Valid! Confirmed after 12m34s.

**Result in 3CxF9Fx50S:** Valid! Confirmed after 15m14s.

**Result in 12CxF12Fx50S:** Valid! Confirmed after 21m20s.
H-13  Except the initial phase, the stock price cannot fall below 180.

In LTL: $G ((StockPrice > 180) \Rightarrow (G StockPrice > 180))$

In ProMeLa: 
```
#define spGt180 (StockPrice > 180)
[](spGt180 -> []spGt180)
```

Result in 6Cx6Fx50S: Valid! Confirmed already in Section 4.4.6.
Result in 9Cx3Fx50S: Valid! Confirmed after 12m22s.
Result in 3Cx9Fx50S: Valid! Confirmed after 15m14s.
Result in 12Cx12Fx50S: Valid! Confirmed after 22m33s.

H-14  The traders in total will never be interested in more than 35 stocks.

In LTL: $G stocksLEt35$

In ProMeLa: 
```
#define stocksLEt35 (Stocks[0]+...+Stocks[11] <= 35)
[](stocksLEt35)
```

Result in 6Cx6Fx50S: Not valid! Error trail found in 1h9m45s.
An error trail was found also for the limit of 40 stocks, but was shown true for 45 stocks. Notice that the long time needed for finding the error already indicates that it is not very likely that such a situation occur.

In LTL: $G stocksLEt45$

In ProMeLa: 
```
#define stocksLEt45 (Stocks[0]+...+Stocks[11] <= 45)
[](stocksLEt45)
```

Result in 6Cx6Fx50S: Valid! Confirmed after 1h11m02s.
Result in 9Cx3Fx50S: Valid! Confirmed after 13m24s.
Result in 3Cx9Fx50S: Valid! Confirmed after 15m01s.

H-21  From some point in time, the average rating of fundamentalists is and remains higher/lower than the average rating of chartists.

In LTL: $F G (avgRatF > avgRatC)$

In ProMeLa: 
```
define avgRatC ((Rating[0]+...+Rating[5])/6)
define ratFGtC (avgRatF > avgRatC)
<>[]ratFGtC
```

Result in 6Cx6Fx50S: Valid! Confirmed already in Section 4.4.6.
Result in 9Cx3Fx50S: Valid! Confirmed after 20m16s.
Result in 3Cx9Fx50S: Valid! Confirmed after 14m58s.
Chapter 4. Experimental Study

In LTL: \( F G (avgRatF < avgRatC) \)

In ProMeLa: #define ratFLtC (avgRatF < avgRatC)
 <>[]ratFLtC

Result in 6Cx6Fx50S: Not valid! Confirmed already in Section 4.4.6.

Result in 9Cx3Fx50S: Not valid! Error trail found in 5.1s.

Result in 3Cx9Fx50S: Not valid! Error trail found in 3.2s.

H-71  Stock price is positioned asymmetrically above the mean value 200.

In LTL: \( G ((Steps = 200) \Rightarrow (G StockPrice > 190)) \)

In ProMeLa: #define stEt200 (Steps == 200)
    #define spGt190 (StockPrice > 190)
    [](stEt200 -> []spGt190)

Result in 6Cx6Fx50S: Not valid! Error trail found in 43m10s.

In LTL: \( G ((Steps = 200) \Rightarrow (G StockPrice > 185)) \)

In ProMeLa: #define spGt185 (StockPrice > 185)
    [](stEt200 -> []spGt185)

Result in 6Cx6Fx50S: Valid! Confirmed after 1h42m40s.

This means that while the price can reach the value 190, it cannot reach 185, so the lower bound of the stock-price curve is somewhere between these two.

In LTL: \( G ((Steps = 200) \Rightarrow (G StockPrice < 225)) \)

In ProMeLa: #define spLt225 (StockPrice < 225)
    [](stEt200 -> []spLt225)

Result in 6Cx6Fx50S: Not valid! Error trail found in 1h40m35s.

Since we have confirmed via H-12 that stock price is globally lower than 230, and this verification reveals that it may reach the value 225, we can conclude that the upper bound is somewhere between 225 and 230. Note that the same result was confirmed also for the limits 210, 215 and 220 before getting to 225.
Chapter 5

Discussion of the Results

"The paradigm of agent-based, behavioral economics, behavioral finance and bounded rationality is rapidly expanding. Heterogeneity is likely to play a key role in this approach, and agent-based computational HAMs deserve high priority in future work. Will an analytical approach survive within more computational oriented research in the 21st century?"


This chapter summarizes the findings encountered within the experimental study, discusses possible model extensions, and concludes with evaluation of the strengths and weaknesses of the automata-based approach applied to modelling and examination of socio-economic systems.

5.1 Results and observations

Within the experimental study, we have constructed 10 main models in a number of variations (see Table 5.1), and performed over a hundred of simulations, from which about 30 simulations were included in the text of the thesis (presented in 84 charts). We have formulated 18 hypotheses (Table 5.2) and verified them on different versions of the models (see Table 5.3 summarizing the main results) within more than a hundred of verifications (selected 104 verification reports included on the attached CD).
Chapter 5. Discussion of the Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
<th>Defined in</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-01</td>
<td>Basic model with simple traders</td>
<td>Section 4.2.3</td>
</tr>
<tr>
<td>M-02</td>
<td>Limited amount of speculative money</td>
<td>Section 4.2.4</td>
</tr>
<tr>
<td>M-03</td>
<td>Model with two types of traders</td>
<td>Section 4.2.7</td>
</tr>
<tr>
<td>M-04</td>
<td>Rating of trader strategies added</td>
<td>Section 4.4.1</td>
</tr>
<tr>
<td>M-05</td>
<td>Negotiation time added</td>
<td>Section 4.4.2</td>
</tr>
<tr>
<td>M-06</td>
<td>Effect of the initial amount of money</td>
<td>Section 4.4.3</td>
</tr>
<tr>
<td>M-07</td>
<td>Stock value changes added</td>
<td>Section 4.4.4</td>
</tr>
<tr>
<td>M-08</td>
<td>Inter-fundamentalist variability added</td>
<td>Section 4.4.5</td>
</tr>
<tr>
<td>M-09</td>
<td>Inter-chartist variability added</td>
<td>Section 4.4.6</td>
</tr>
<tr>
<td>M-10</td>
<td>Different proportions of traders</td>
<td>Section 4.4.7</td>
</tr>
</tbody>
</table>

Table 5.1: Outline of the examined models

**Models.** We started with an initial model of the market inspired by Zeeman [19], model M-03, which was constructed in an iterative way (with models M-01 and M-02 created along the process). After validating and analysing the model, we have extended the model with rating of trading strategies (model M-04) and started to study effect of various modifications of the model. These included incorporation of some negotiation time (model M-05), changing initial amount of speculative money (model M-06), and evolving fundamental value of the traded stocks (model M-07). The most important modification (model M-08 and M-09) introduced variability within the groups of fundamentalists and chartists and studied its effect. The final model (model M-10) was used to examine different proportions of fundamentalists and chartists on the market.

**General observations.** In summary, we can confirm the observations of other authors [6], describing fundamentalists as a stabilizing force, pushing the stock price in the directions of stock fundamental value, and chartists as a destabilizing force, pushing the price away from stock value. When the proportion of chartists exceeds some critical value, the chartists’ belief in the price trend becomes self-fulfilling, causing the price to deviate from the fundamental value.

**Convergence of the market.** When concerned with the initial model of the market (model M-03 defined in Section 4.2.7), we observed very fast convergence of the market to a stable state characterized with the equilibrium stock price, equal to the fundamental value. After introduction of a number of non-regularities to the
<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>H-01</td>
<td>The model is deadlock-free.</td>
</tr>
<tr>
<td>H-02</td>
<td>The stock price is always above the initial value.</td>
</tr>
<tr>
<td>H-03</td>
<td>The stock price is always below the maximal value.</td>
</tr>
<tr>
<td>H-04</td>
<td>The money amount of a trader cannot fall to zero or below.</td>
</tr>
<tr>
<td>H-11</td>
<td>The stock price always converges to the value 200.</td>
</tr>
<tr>
<td>H-12</td>
<td>The stock price is always lower than 230.</td>
</tr>
<tr>
<td>H-13</td>
<td>Except the initial phase, the stock price cannot fall below 180.</td>
</tr>
<tr>
<td>H-14</td>
<td>The traders in total will never be interested in more than $k$ stocks.</td>
</tr>
<tr>
<td>H-21</td>
<td>From some point in time, the average rating of fundamentalists is and remains (a) higher / (b) lower than the average rating of chartists.</td>
</tr>
<tr>
<td>H-22</td>
<td>The inter-fundamentalist variability is higher/lower than the inter-chartist variability. Can it happen that two traders of the same type at one point differ in 4 stocks, or 50 points in rating?</td>
</tr>
<tr>
<td>H-31</td>
<td>We want to check what initial amount of speculative money is needed so that momentary money held by a trader never falls below 200.</td>
</tr>
<tr>
<td>H-32</td>
<td>We want to check what initial amount of speculative money is needed so that momentary money held by a fundamentalist never falls below 200.</td>
</tr>
<tr>
<td>H-33</td>
<td>We want to check what initial amount of speculative money is needed so that momentary money held by a chartist never falls below 200.</td>
</tr>
<tr>
<td>H-41</td>
<td>The stock price converges to the stock value.</td>
</tr>
<tr>
<td>H-51</td>
<td>Fundamentalists with the stock value closer to 200 have better rating.</td>
</tr>
<tr>
<td>H-61</td>
<td>The ratings of the chartist strategies at the end always confirm that strategy 0 is better than strategy 2.</td>
</tr>
<tr>
<td>H-62</td>
<td>The ratings of the chartist strategies at the end always confirm that strategy 0 is worse than strategy 2.</td>
</tr>
<tr>
<td>H-71</td>
<td>Stock price is positioned asymmetrically above the mean value 200.</td>
</tr>
</tbody>
</table>

Table 5.2: Defined and verified hypotheses
### Chapter 5. Discussion of the Results

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Verification result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H-01</td>
<td>Valid on M-01 – M-10</td>
</tr>
<tr>
<td>H-02</td>
<td>Valid on M-01 – M-10</td>
</tr>
<tr>
<td>H-03</td>
<td>Valid on M-01 – M-10</td>
</tr>
<tr>
<td>H-04</td>
<td>Falling to zero possible, below zero impossible on M-02</td>
</tr>
<tr>
<td>H-11</td>
<td>Valid on M-03 in case of 6Cx6Fx50S and 2Cx6Fx50S, and M-04 – M-05; invalid on M-03 in case of 6Cx2Fx50S, and M-08 – M-10</td>
</tr>
<tr>
<td>H-12</td>
<td>Valid on M-03 in case of 6Cx6Fx50S and 3Cx9Fx50S, M-04 – M-05, and M-08 – M-10; invalid on M-03 in case of 9Cx3Fx50S</td>
</tr>
<tr>
<td>H-13</td>
<td>Valid on M-03 – M-04 and M-08 – M-10; invalid on M-05 (the limit of 175 needed)</td>
</tr>
<tr>
<td>H-14</td>
<td>Valid on M-03 – M-04 when ( k = 35 ), on M-05 when ( k = 40 ), and M-10 when ( k = 45 ); invalid on M-03 – M-04 when ( k = 30 ), on M-05 when ( k = 35 ), and M-10 when ( k = 40 )</td>
</tr>
<tr>
<td>H-21</td>
<td>Valid (a) on M-08 – M-10; invalid (a) on M-04; invalid (b) on M-04 and M-08 – M-10</td>
</tr>
<tr>
<td>H-22</td>
<td>On M-04 valid for 50 stocks among both F and C (can differ in 50 stocks); invalid for 60 stocks among both F and C; valid for 4 stocks among F; invalid for 4 stocks among C</td>
</tr>
<tr>
<td>H-31</td>
<td>On M-06 amount ( M = 2000 ) insufficient, ( M = 2200 ) sufficient</td>
</tr>
<tr>
<td>H-32</td>
<td>On M-06 amount ( M = 1000 ) insufficient, ( M = 1200 ) sufficient</td>
</tr>
<tr>
<td>H-33</td>
<td>On M-06 amount ( M = 2000 ) insufficient, ( M = 2200 ) sufficient</td>
</tr>
<tr>
<td>H-41</td>
<td>Valid on M-07</td>
</tr>
<tr>
<td>H-51</td>
<td>Valid on M-08 in case of both normal and uniform distribution</td>
</tr>
<tr>
<td>H-61</td>
<td>In most cases valid on M-09, more than 1 hour needed for finding a counterexample</td>
</tr>
<tr>
<td>H-62</td>
<td>Invalid on M-09, 6.1 seconds needed for finding a counterexample</td>
</tr>
<tr>
<td>H-71</td>
<td>Valid on M-10, the lower bound reported between 185–190, the upper bound reported within 225–230</td>
</tr>
</tbody>
</table>

Table 5.3: Summary of the verification results
model (like negotiation time, evolving fundamental value, etc.), the time needed to reach the stable state became longer. The length of the convergence process was shown to be influenced also by the proportions of the traders, and their internal variability. In any case, however, the convergence tendencies remained obvious.

**Amplitude of the stock price curve.** The main indicator of the market convergence in our model is the curve of the stock-price evolution, and its proximity to the fundamental stock value. We have shown that if the number of emitted stocks on the market and the amount of speculative money of the traders are somewhat reasonable (surprisingly not very high\(^1\), the stock-price curve is independent on these and is rather sensitive to the strategies of the traders, and their proportions on the market. We could see that the amplitude is lower (curve closer to the fundamental value) whenever the number of fundamentalists outbalances, and higher otherwise.

**Position of the stock-price curve.** Whenever all fundamentalists assume the same stock value, the stock-price curve is positioned around, and converging to this fundamental value. The situation changes when we introduce some uncertainty to the fundamentalists—via assigning them different values from a (normal/uniform) distribution with the mean value equal to the former fundamental value. Under such conditions, the price can converge to a different value (possibly quite far from the mean).

There is one more surprising observation regarding the position of the stock-value curve. In Section 4.4.7, we could notice that despite the symmetry of both fundamentalists values and trading rules of fundamentalists and chartists, the stock price have a strong tendency to be positioned above the mean value. We have, however, no clear explanation for that.

**Ratings of the traders.** Along the experimental study, one of our main concerns was about the profit of different trading strategies, defined as strategies rating. In case of the initial models, with fast convergence of the market, we could not observe any significant deviation of the strategies ratings. But after introducing some non-regularities to the model, the convergence phase was prolonged, and we could witness a strong deviation of ratings, in favour of fundamentalists. The advantage of fundamentalists is surprisingly present under all examined modifications of the model, including changing fundamental value, weakening fundamentalists knowledge

\(^1\)Notice that Stocks=50 and M=1000 used in our study was far above such a limit, even when the number of traders was around 20, and so the traders could in total demand about 100 stocks.
about the real stock value, or changing the proportions of traders. The only condition needed for it to show up, is the prolonged convergence phase. Note that we would not learn about this phenomenon if we did not introduce the constructs that keep the market a bit longer out of equilibrium.

Besides the rating of fundamentalists comparing to chartists, we have also studied the rating of individual strategies within each group. We discovered and formally confirmed that among fundamentalists, the best rating can be found at fundamentalists with the stock value closer to the mean fundamental value, and among chartists, the best strategy is the careful one, i.e. following only such trends that are unambiguously confirmed by all steps in the observed history. The second finding may be surprising, because chartists with such a strategy have the longest delay behind fundamentalists. On the other hand, this behaviour protects them from the losses during the last phase of price oscillation before convergence.

Variability among traders. There are two kinds of variability among traders. First, the defined variability, both between the two trader types, and among traders of the same type following different variations of the strategies. Second, the model-inherent variability emerging from the non-determinism of the model between traders following (by definition) the same strategies. In our study, we were concerned with both types. Regarding the defined variability, we have examined for instance the difference in the ratings of different strategies, or in the upper limit on the amount of initial speculative money that no more limits traders behaviour. Regarding the model-inherent variability, we saw that even in case of two traders following the same strategy, we can find a significant difference in, for instance, actual number of owned stocks or rating of the trader.

Verification experience. In most of the cases, the full-exploration verification technique of the employed SPIN tool (see Section 3.4.1) was unable to tackle the full model (on a personal computer used to analyse the model). Hence for the verification, we employed the BitState approximation technique implemented within the SPIN tool. Our experience shows that the technique has a very good coverage of the state-space, and while confirming of a correct property needed often tens of minutes (sometimes hours), in case of invalid properties (when the technique searches for a counterexample in the model) was such a counterexample reported typically within seconds. Hence even if we do not have the full certainty about the validity of the result, if no counterexample was found within an hour or so, this gives us a fair evidence that the property is valid.
5.2 Possible extensions of the model

There is a number of possible extensions of the model that could bring interesting results. We outline the major ones that we have discovered.

Policy-oriented modifications. The model of the market could be used to evaluate the effect of various policy modifications, like external regulations of the evolution, charging of the purchase/sale operations (especially if defined in an asymmetric way), or tax introduction.

Local interaction. The current model does not consider mutual influence among the traders. It could be interesting to extend the model with non-deterministic local interactions, allowing the traders for instance to pass their strategy to some traders in their proximity.

Survival conditions. Another natural condition, easily representable in terms of finite automata, is the dynamic evolution of the market (its trading subjects). If included, traders with poor rating could be forced to leave the market, or change their strategy, traders with very good rating could motivate new traders to enter the market and copy their strategy.

One-to-one trading. Under the current settings, traders can buy a traded product (stock), whenever they have enough money, and sell whenever they have anything to sell. It would be interesting to study also the markets where each seller first needs to find an interested buyer (and vice versa) before the trading can take place. The searching could include the process of price negotiation.

5.3 Strengths and weaknesses of the approach

What lessons have we learnt during the work on this thesis, and the experimental study in particular? What are the strengths and weaknesses (S&W) of the agent-based approach to economic analysis, and of the automata theory to realize its principles?
5.3.1 S&W of the agent-based paradigm

The essential principle of the agent-based paradigm is the building of economic models out of simple interacting agents, following usually very regular rules. As distinct to traditional economics, this allows the models to consist of a large number of agents, with high variability between individuals. This two aspects, inherent also to real economic system, can then give rise to the complex phenomena common in reality.

The critics of this approach to economic modelling point out mainly the bounded rationality of individual agents. Due to the simplicity of the behavioural rules that agents typically follow, the critics do not really believe in the validity of the phenomena observed when analysing such models. Another disturbance of the results is seen in the inherent non-determinism of agent-based models, which implies the danger of reinforced disturbing effects propagated from the micro to the macro level.

On the other hand, the advocates of the agent-based approach oppose with the argument that bounded rationality and high level of non-determinism are more realistic and natural to economic subjects than full rationality and deterministic behaviour. This discussion has not come to a clear conclusion yet.

5.3.2 S&W of the automata theory in agent-based modelling

We have seen that finite automata represent very intuitive models of agent lifecycles, naturally including guard conditions on their behaviour and constructs for interaction with other agents. The models can be both fully decentralized, and centralized (when a central coordination agent is introduced), and can consist of a large number of interacting agents defined parametrically, and hence different from each other whenever appropriate. Automata-based models have the advantage that they can be created on a micro level, modelling each interacting entity with a simple automaton, and the macro models can be generated and examined automatically with existing state-space exploration tools. Thanks to the computational simplicity of the formalism, the models can consist of a very large number of agents\(^2\). The tools implement both simulation of the models (helping to uncover new phenomena emerging from the interaction) and formal verification (helping to prove hypotheses about observed behaviour).

\(^2\)Note that the number of agents included in our experimental study was limited by the computational capacity of the personal computer used for the analysis.
On the other hand, due to the simplicity of the formalism, this approach is not well suited for the models dependent on sophisticated computations performed by agents. Even if simple computations are possible, the restriction of dealing with only natural numbers is already quite limiting, and may have tricky consequences.

At the same time, the shift from equation-based to automata-based modelling is not a trivial task for a modeller. Construction of valid automata-based models requires a certain level of understanding of the formalism. A modeller needs to acquire a different way of thinking about the agents, and gain the insight into the effects caused by model constructs, i.e. the semantics of model composition. However, this task can be significantly simplified by providing the modeller with a number of examples and modelling guidelines, in a fashion given by this thesis.
Chapter 5. Discussion of the Results
Chapter 6

Conclusion

“Humans often make mistakes in choosing too complicated formalisms to express what they want. But it is just because they are not able to accept that much simple formalisms are capable of the same, and what more, can do it more clearly.”

Stephen Wolfram in A New Kind of Science [18]

Economics and finance are witnessing an important paradigm shift, from a representative, rational agent approach towards a behavioural, agent-based approach in which markets are populated with boundedly rational, heterogeneous agents using rule of thumb strategies [6]. In physics, biology, informatics, and other kinds of science, models of heterogeneous interacting systems use to be successfully constructed on the basis of the automata theory. The lifecycle of each agent is described with a simple automaton, and the full system defined as a composition of these.

In economics, however, not much work has been done in this domain. Despite the natural connection of the agent-based economics to the principles of automata-based reasoning, agent-based models use to be constructed with different means, typically as systems of linear and non-linear equations.

This thesis contributes to the current state of the art in socio-economic system modelling via practically exploring the capability and limits of the automata theory in the domain of agent-based modelling. After outlining existing results in the field and defining an appropriate research methodology, the thesis presents an extensive experimental study. In the study, we guide the reader through the process of specifying (with an appropriate level of abstraction), modelling, validating, simulating
Chapter 6. Conclusion

and formally analysing a selected economic model, a fundamentalist/chartist stock market.

Besides the primary contribution, which is the demonstration of the application process, we present a number of findings about the model, and summarize them in a separate chapter including also discussion of the observed strengths and weaknesses of the approach.

6.1 Possible future directions

In Chapter 5, we have outlined a number of possible upgrades of the examined fundamentalist/chartist model. The natural direction would be then to include such extensions in the model. One should however bear in mind that the intended resulting model is not the most complex one, but quite the opposite. Model upgrades should be added only to the extent when the model starts to exhibit a new, interesting behaviour. After that, all the complexities in the model that have no effect on the phenomenon should be filtered away. The result should be again a simple model on the micro level, which however captures the emerging phenomena at the aggregate level, so that their grounds can be easily traced back to the underlying rules [6].
Bibliography


Online resources


Appendix A

Additional Data

A.1 Automata
Appendix A. Additional Data

Figure A.1: Model of a chartist with trend update
Appendix A. Additional Data

Figure A.2: Model of a fundamentalist with trend update
Figure A.3: Model of a fundamentalist with end states
Figure A.4: Model of a chartist with end states
Appendix A. Additional Data

A.2 Charts

A.2.1 Limited amount of speculative money

Figure A.6: Number of stocks held by the traders for the 2Tx10S settings
Appendix A. Additional Data

A.2.2 Model of a stock market with both types of traders

Figure A.7: Number of stocks held by the traders for the 2Tx15S settings

Figure A.8: Number of stocks held by the traders for the 2Cx2Fx50S settings
Appendix A. Additional Data

A.2.3 Initial analysis of the model

Figure A.9: Number of stocks held by fundamentalists for the 6Cx6Fx50S settings

Figure A.10: Number of stocks held by chartists for the 6Cx6Fx50S settings
Appendix A. Additional Data

Figure A.11: Number of stocks held by individual traders for the 2Cx6Fx50S settings

Figure A.12: Number of stocks held by individual traders for the 6Cx2Fx50S settings
Appendix A. Additional Data

A.2.4 Strategies evaluation

Figure A.13: Profit of fundamentalists for the 6Cx6Fx50S settings

Figure A.14: Profit of chartists for the 6Cx6Fx50S settings
Figure A.15: Rating of fundamentalists for the 6C6Fx50S settings

Figure A.16: Rating of chartists for the 6C6Fx50S settings
Appendix A. Additional Data

A.2.5 Influence of the negotiation time

Figure A.17: Amount of speculative money in both groups—evolving price

Figure A.18: Amount of speculative money in both groups—fixed price
A.2.6 Influence of the initial amount of speculative money

Figure A.19: Amount of speculative money of selected individual traders (for M=550)

Figure A.20: Amount of speculative money in both groups (for M=550)
Appendix A. Additional Data

A.2.7 Effect of inter-fundamentalist variability

Figure A.21: Number of stocks held by individual fundamentalists (simulation A)

Figure A.22: Number of stocks held by individual fundamentalists (simulation B)
Figure A.23: Number of stocks held by individual fundamentalists (simulation C)
Appendix A. Additional Data

A.2.8 Effect of inter-chartist variability

Figure A.24: Rating of the traders with respect to their strategies (model 001122, alternative simulation I.)

Figure A.25: Rating of the traders with respect to their strategies (model 001122, alternative simulation II.)
Figure A.26: Rating of the traders with respect to their strategies in long run (model 001122, alternative simulation I.)

Figure A.27: Rating of the traders with respect to their strategies in long run (model 001122, alternative simulation II.)
Appendix A. Additional Data

A.2.9 Different proportions of the two types of traders

Figure A.28: Number of stocks held by all the traders for the 15Cx6Fx50S settings

Figure A.29: Number of stocks held by all the traders for the 12Cx12Fx50S settings
Appendix B

ProMeLa Models

B.1 Basic model with simple traders

mtype { buy, bought, sell, sold };  
chan market = [0] of { mtype, byte };  
short StockPrice = 150; /* Stock price and its initialisation */

active[2] proctype T () /* Traders */
{
  do
    :: market!buy(_pid); /* Attempt for buying */
    atomic /* Buying processed */
    {
      market?bought(eval(_pid)) ->
      StockPrice = StockPrice + 2;
    }
    :: market!sell(_pid); /* Attempt for selling */
    atomic /* Selling processed */
    {
      market?sold(eval(_pid)) ->
      StockPrice = StockPrice - 2;
    }
  od
}

active[2] proctype S () /* Stocks */
{
  byte subj;

  do
    :: atomic /* Being bought */
B.2 Traders with a limited amount of speculative money

mtype { buy, bought, sell, sold };  
chan market = [0] of { mtype, byte };  
#define M 1000; /* Initial amount of speculative money */  
short StockPrice = 150; /* Stock price and its initialisation */  
byte Stocks[2]; /* Array for keeping the number of stocks held by each trader */  

active[2] proctype T () /* Traders */  
{  
short Money = M;  

do :: atomic /* Attempt for buying */  
{  
Money > StockPrice ->  
market!buy(_pid);  
}  
atomic /* Buying processed */  
{  
market?bought(eval(_pid)) ->  
Stocks[_pid]++;  
Money = Money - (StockPrice + 1);  
StockPrice = StockPrice + 2;  
}  
:: atomic /* Attempt for selling */  
{  
Stocks[_pid] > 0 ->  
market!sell(_pid);  
}  
atomic /* Selling processed */
Appendix B. ProMeLa Models

{  
  market?sold(eval(_pid)) ->
  Stocks[pid]--;
  Money = Money + (StockPrice - 1);
  StockPrice = StockPrice - 2;
}
od
}

active[12] proctype S () /* Stocks */
{
  byte subj;
  do
    :: atomic /* Being bought */
    {
      market?buy(subj);
      market!bought(subj);
    };
  atomic /* Being sold */
  {
    market?sell(eval(subj));
    market!sold(subj);
  }
  od
}

B.3 Model of a stock market with two types of traders

mtype { buy, bought, sell, sold };
chan market = [0] of { mtype, byte };
#define M 1000; /* Initial amount of speculative money */
short StockPrice = 180; /* Stock price and its initialisation */
byte PriceTrend = 1; /* Price trend and its initialisation to a neutral value */
byte PT[5]; /* Auxiliary array for keeping the history of the price trend */
#define T 4; /* Number of traders */
byte Stocks[T]; /* Array for keeping the number of stocks held by each trader */
bit go = 0;

active[2] proctype F () /* Fundamentalists */
{
  short Money = M;
  byte StockValue = 200;
}
Appendix B. ProMeLa Models

byte temp;
go == 1;
end:

do
:: atomic /* Attempt for buying */
{
    (StockPrice < StockValue) && (Money > StockPrice) ->
    market!buy(_pid);
}
atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->
    Stocks[_pid]++;
    Money = Money - (StockPrice + 1);
    StockPrice = StockPrice + 2;
    /* PriceTrend update */
    if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
:: atomic /* Attempt for selling */
{
    (Stocks[_pid] > 0) && (StockPrice > StockValue) ->
    market!sell(_pid);
}
atomic /* Selling processed */
{
    market?sold(eval(_pid)) ->
    Stocks[_pid]--;
    Money = Money + (StockPrice - 1);
    StockPrice = StockPrice - 2;
    /* PriceTrend update */
    if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
fi;
Appendix B. ProMeLa Models

:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
}

active[2] proctype C () /* Chartists */
{
  short Money = M;
  byte temp;
  go == 1;
}

end:

do
:: atomic /* Attempt for buying */
{
  (PriceTrend == 2) && (Money > StockPrice) ->
    market!buy(.pid);
}
atomic /* Buying processed */
{
  market?bought(eval(.pid)) ->
    Stocks[.pid]++;
    Money = Money - (StockPrice + 1);
    StockPrice = StockPrice + 2;

  /* PriceTrend update */
  if
    :: temp == 0 -> PriceTrend = 0;
    :: temp == 1 -> PriceTrend = 0;
    :: temp == 2 -> PriceTrend = 1;
    :: temp == 3 -> PriceTrend = 1;
    :: temp == 4 -> PriceTrend = 2;
    :: temp == 5 -> PriceTrend = 2;
  fi;
}
:: atomic /* Attempt for selling */
{
  (Stocks[.pid] > 0) && (PriceTrend == 0) ->
    market!sell(.pid);
}
atomic /* Selling processed */
{
  market?sold(eval(.pid)) ->
    Stocks[.pid]--;
}
Appendix B. ProMeLa Models

Money = Money + (StockPrice - 1);
StockPrice = StockPrice - 2;

/* PriceTrend update */
if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
}

active[25] proctype S () /* Stocks */
{
byte subj;

di
:: atomic /* Being bought */
{
market?buy(subj);
market!bought(subj);
};

di
:: atomic /* Being sold */
{
market?sell(eval(subj));
market!sold(subj);
}

init
{
/* PT initialization */
PT[0]=0; PT[1]=0;

go = 1;
}
B.4 Rating of trader strategies added

mtype { buy, bought, sell, sold };  
chan market = [0] of { mtype, byte };

#define M 1000; /* Initial amount of speculative money */
short StockPrice = 150; /* Stock price and its initialisation */
byte PriceTrend = 1; /* Price trend and its initialisation to a neutral value */
byte PT[5]; /* Auxiliary array for keeping the history of the price trend */

#define T 12; /* Number of traders */
byte Stocks[12]; /* Array for keeping the number of stocks held by each trader */
short Rating[12]; /* Array for keeping the rating of each trader */

bit go = 0;

active[6] proctype F () /* Fundamentalists */
{
  short Money = M;
  byte StockValue = 200;
  byte trader, temp;
  go == 1;
end:

do :: atomic /* Attempt for buying */
{
  (StockPrice < StockValue) && (Money > StockPrice) ->
    market!buy(_pid);
}
atomic /* Buying processed */
{
  market?bought(eval(_pid)) ->
    do /* Rating update */
      :: trader < T ->
        Rating[trader] = Rating[trader] + Stocks[trader];
        trader++;
      :: else -> trader=0; break;
    od;

    Stocks[_pid]++;
    Money = Money - (StockPrice + 1);
    StockPrice = StockPrice + 2;

    /* PriceTrend update */
    if
Appendix B. ProMeLa Models

:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;

:: atomic /* Attempt for selling */
{
    (Stocks[_pid] > 0) && (StockPrice > StockValue) ->
    market!sell(_pid);
}

atomic /* Selling processed */
{
    market?sold(eval(_pid)) ->
    Stocks[_pid]--;
    Money = Money + (StockPrice - 1);
    StockPrice = StockPrice - 2;

do /* Rating update */
:: trader < T ->
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
:: else -> trader=0; break;
od;

/* PriceTrend update */
if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;

active[6] proctype C () /* Chartists */
{
    short Money = M;
    byte trader, temp;
    go == 1;
}

136
APPENDIX B. PROMLEA MODELS

do :: atomic /* Attempt for buying */
{
    (PriceTrend == 2) && (Money > StockPrice) ->
    market!buy(_pid);
}
atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->

do /* Rating update */
:: trader < T ->
    Rating[trader] = Rating[trader] + Stocks[trader];
    trader++;
:: else -> trader=0; break;
od;

Stocks[_pid]++;
Money = Money - (StockPrice + 1);
StockPrice = StockPrice + 2;

/* PriceTrend update */
if :: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
:: atomic /* Attempt for selling */
{
    (Stocks[_pid] > 0) && (PriceTrend == 0) ->
    market!sell(_pid);
}
atomic /* Selling processed */
{
    market?sold(eval(_pid)) ->

Stocks[_pid]--;
Money = Money + (StockPrice - 1);
StockPrice = StockPrice - 2;

do /* Rating update */
:: trader < T ->
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
:: else -> trader=0; break;
od;
Appendix B. ProMeLa Models

/* PriceTrend update */
if 
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
}

active[50] proctype S () /* Stocks */
{
byte subj;

endBuy:

do 
:: atomic /* Being bought */
{
market?buy(subj);
market!bought(subj);
};
endSell:

atomic /* Being sold */
{
market?sell(eval(subj));
market!sold(subj);
};

init
{
/* PT initialisation */
PT[0]=0; PT[1]=0;

go = 1;
}
### B.5 Negotiation time added

```pro meladsl
mtype { buy, bought, sell, sold }; chan market = [0] of { mtype, byte }; #define M 1000; /* Initial amount of speculative money */ short StockPrice = 150; /* Stock price and its initialisation */ byte PriceTrend = 1; /* Price trend and its initialisation to a neutral value */ byte PT[5]; /* Auxiliary array for keeping the history of the price trend */ #define T 12; /* Number of traders */ byte Stocks[12]; /* Array for keeping the number of stocks held by each trader */ short Rating[12]; /* Array for keeping the rating of each trader */ bit go = 0;

active[6] proctype F () /* Fundamentalists */ { short Money = M; byte StockValue = 200; short pay, obtain; byte trader, temp; go == 1; end:
```
Appendix B. ProMeLa Models

```promla
appendix

:: atomic /* Attempt for selling */
{
  (Stocks[_pid] > 0) && (StockPrice > StockValue) ->
  obtain = StockPrice - 1;
  market!sell(_pid);
}
atomic /* Selling processed */
{
  market?sold(eval(_pid)) ->
  Stocks[_pid]--;
  Money = Money + obtain;
  StockPrice = StockPrice - 2;

do /* Rating update */
  :: trader < T ->
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
  :: else -> trader=0; break;
  od;

/* PriceTrend update */
if
  :: temp == 0 -> PriceTrend = 0;
  :: temp == 1 -> PriceTrend = 0;
  :: temp == 2 -> PriceTrend = 1;
  :: temp == 3 -> PriceTrend = 1;
  :: temp == 4 -> PriceTrend = 2;
  :: temp == 5 -> PriceTrend = 2;
fi;
}
o
d

active[6] proctype C () /* Chartists */
{
  short Money = M;
  short pay, obtain;
  byte trader, temp;
}
```

Appendix B. ProMeLa Models

go == 1;
end:

do :: atomic /* Attempt for buying */
{
    (PriceTrend == 2) && (Money > StockPrice) ->
        pay = StockPrice + 1;
        market!buy(_pid);
}
atomic /* Buying processed */
{
    atomic /* Attempt for selling */
    {
        (Stocks[_pid] > 0) && (PriceTrend == 0) ->
            obtain = StockPrice - 1;
            market!sell(_pid);
    }
    atomic /* Selling processed */
    {
        market?sold(eval(_pid)) ->
            Stocks[_pid] =;
            Money = Money + obtain;
            StockPrice = StockPrice - 2;
    }
Appendix B. ProMeLa Models

do /* Rating update */
   :: trader < T ->
      Rating[trader] = Rating[trader] - Stocks[trader];
      trader++;%
   :: else -> trader=0; break;
   od;
/* PriceTrend update */
   if
      :: temp == 0 -> PriceTrend = 0;
      :: temp == 1 -> PriceTrend = 0;
      :: temp == 2 -> PriceTrend = 1;
      :: temp == 3 -> PriceTrend = 1;
      :: temp == 4 -> PriceTrend = 2;
      :: temp == 5 -> PriceTrend = 2;
   fi;
   od

active[50] proctype S () /* Stocks */
{
   byte subj;

   endBuy:

   do
      :: market?buy(subj); /* Being bought */
      market!bought(subj);
   od

   endSell:

      market?sell(eval(subj)); /* Being sold */
      market!sold(subj);
   od

   init
   {
      /* PT initialisation */
      PT[0]=0; PT[1]=0;
      go = 1;
   }
B.6 Stock value changes added

mtype { buy, bought, sell, sold };  
chan market = [0] of { mtype, byte };  
#define M 1000; /* Initial amount of speculative money */  
short StockPrice = 150; /* Stock price and its initialisation */  
byte PriceTrend = 1; /* Price trend and its initialisation to a neutral value */  
byte PT[5]; /* Auxiliary array for keeping the history of the price trend */  
byte StockValue = 200; /* Global StockValue for all fundamentalists */  
byte PriceChange = 0; /* Auxiliary variable helping us to know when the price change is expected */  
#define T 12; /* Number of traders */  
byte Stocks[12]; /* Array for keeping the number of stocks held by each trader */  
short Rating[12]; /* Array for keeping the rating of each trader */  
bit go = 0;  

active[6] proctype F () /* Fundamentalists */  
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;
    go == 1;
}

end:

do
:: atomic /* Attempt for buying */
{
    (StockPrice < StockValue) && (Money > StockPrice) ->
        pay = StockPrice + 1;
        market!buy(_pid);
}
atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->
        do /* Rating update */
            :: trader < T ->
                Rating[trader] = Rating[trader] + Stocks[trader];
                trader++;
            :: else -> trader=0; break;
            od;
        Stocks[_pid]++;
        Money = Money - pay;
        StockPrice = StockPrice + 2;
/* PriceTrend update */

APPENDIX B. ProMeLa Models

if
  :: temp == 0 -> PriceTrend = 0;
  :: temp == 1 -> PriceTrend = 0;
  :: temp == 2 -> PriceTrend = 1;
  :: temp == 3 -> PriceTrend = 1;
  :: temp == 4 -> PriceTrend = 2;
  :: temp == 5 -> PriceTrend = 2;
fi;

/* Price change */
PriceChange++;
if
  :: PriceChange == 10 -> StockValue++; PriceChange = 0;
  :: else -> skip;
fi;
}
:: atomic  /* Attempt for selling */
{
  (Stocks[_pid] > 0) && (StockPrice > StockValue) ->
  obtain = StockPrice - 1;
  market!sell(_pid);
}
atomic  /* Selling processed */
{
  market?sold(eval(_pid)) ->

  Stocks[_pid]--;
  Money = Money + obtain;
  StockPrice = StockPrice - 2;

do /* Rating update */
  :: trader < T ->
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
  :: else -> trader=0; break;
od;

/* PriceTrend update */
if
  :: temp == 0 -> PriceTrend = 0;
  :: temp == 1 -> PriceTrend = 0;
  :: temp == 2 -> PriceTrend = 1;
  :: temp == 3 -> PriceTrend = 1;
  :: temp == 4 -> PriceTrend = 2;
  :: temp == 5 -> PriceTrend = 2;
fi;

/* Price change */
PriceChange++;
if
:: PriceChange == 10 -> StockValue++; PriceChange = 0;
:: else -> skip;
fi;
}
}
}

active[6] proctype C () /* Chartists */
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;
    go == 1;
end:

do
:: atomic /* Attempt for buying */
{
    (PriceTrend == 2) && (Money > StockPrice) ->
        pay = StockPrice + 1;
        market!buy(_pid);
    }
atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->

do /* Rating update */
:: trader < T ->
        Rating[trader] = Rating[trader] + Stocks[trader];
        trader++;
:: else -> trader=0; break;
od;

Stocks[_pid]++;
Money = Money - pay;
StockPrice = StockPrice + 2;

/* PriceTrend update */
if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
f1;
Appendix B. ProMeLa Models

/* Price change */
PriceChange++;
if
  :: PriceChange == 10 \rightarrow StockValue++; PriceChange = 0;
  :: else \rightarrow skip;
fi;
}
:: atomic /* Attempt for selling */
{
  (Stocks[_pid] > 0) && (PriceTrend == 0) \rightarrow
  obtain = StockPrice - 1;
  market!sell(_pid);
}
atomic /* Selling processed */
{
  market?sold(eval(_pid)) \rightarrow
  Stocks[_pid]--;
  Money = Money + obtain;
  StockPrice = StockPrice - 2;

do /* Rating update */
  :: trader < T \rightarrow
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
  :: else \rightarrow trader=0; break;
od;

/* PriceTrend update */
if
  :: temp == 0 \rightarrow PriceTrend = 0;
  :: temp == 1 \rightarrow PriceTrend = 0;
  :: temp == 2 \rightarrow PriceTrend = 1;
  :: temp == 3 \rightarrow PriceTrend = 1;
  :: temp == 4 \rightarrow PriceTrend = 2;
  :: temp == 5 \rightarrow PriceTrend = 2;
fi;

/* Price change */
PriceChange++;
if
  :: PriceChange == 10 \rightarrow StockValue++; PriceChange = 0;
  :: else \rightarrow skip;
fi;
}
od
Appendix B. ProMeLa Models

```promla
active[50] proctype S () /* Stocks */
{
    byte subj;

    endBuy:
    do
        :: market?buy(subj);       /* Being bought */
        market!bought(subj);
    endSell:
        market?sell(eval(subj));   /* Being sold */
        market!sold(subj);
    od
}

init
{
    /* PT initialisation */
    PT[0]=0; PT[1]=0;

    go = 1;
}

B.7 Inter-fundamentalist variability added

mtype { buy, bought, sell, sold }; chan market = [0] of { mtype, byte };

#define M 1000;           /* Initial amount of speculative money */
short StockPrice = 150;  /* Stock price and its initialisation */
byte PriceTrend = 1;      /* Price trend and its initialisation to a neutral value */
byte PT[5];              /* Auxiliary array for keeping the history of the price trend */

#define T 12;              /* Number of traders */
byte Stocks[12];         /* Array for keeping the number of stocks held by each trader */
short Rating[12];        /* Array for keeping the rating of each trader */

bit go = 0;

proctype F (byte id; byte StockValue) /* Fundamentalists */
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;

    // Fundamentalist logic...
}
```
Appendix B. ProMeLa Models

\[
\text{go} \equiv 1;
\]

\[
\text{end:}
\]

\[
\text{do}
\]

:: atomic /* Attempt for buying */

{ 
(StockPrice < StockValue) && (Money > StockPrice) ->
  pay = StockPrice + 1;
  market!buy(_pid);
}

atomic /* Buying processed */

{ 
  market?bought(eval(_pid)) ->

    do /* Rating update */
    :: trader < T ->
      Rating[trader] = Rating[trader] + Stocks[trader];
      trader++;
    :: else -> trader=0; break;
    od;

  Stocks[id]++;
  Money = Money - pay;
  StockPrice = StockPrice + 2;

  /* PriceTrend update */
  if
    :: temp == 0 -> PriceTrend = 0;
    :: temp == 1 -> PriceTrend = 0;
    :: temp == 2 -> PriceTrend = 1;
    :: temp == 3 -> PriceTrend = 1;
    :: temp == 4 -> PriceTrend = 2;
    :: temp == 5 -> PriceTrend = 2;
  fi;

  :: atomic /* Attempt for selling */

  { 
    (Stocks[id] > 0) && (StockPrice > StockValue) ->
      obtain = StockPrice - 1;
      market!sell(_pid);
  }

atomic /* Selling processed */

{ 
  market?sold(eval(_pid)) ->

    Stocks[id]--;
    Money = Money + obtain;
    StockPrice = StockPrice - 2;
}
do /* Rating update */
:: trader < T ->
   Rating[trader] = Rating[trader] - Stocks[trader];
   trader++;
:: else -> trader=0; break;
od;

/* PriceTrend update */
if
:: temp == 0 -> PriceTrend = 0;
:: temp == 1 -> PriceTrend = 0;
:: temp == 2 -> PriceTrend = 1;
:: temp == 3 -> PriceTrend = 1;
:: temp == 4 -> PriceTrend = 2;
:: temp == 5 -> PriceTrend = 2;
fi;
}
od
}

proctype C (byte id) /* Chartists */
{
  short Money = M;
  short pay, obtain;
  byte trader, temp;
  go == 1;
end:

do
:: atomic /* Attempt for buying */
{
  (PriceTrend == 2) && (Money > StockPrice) ->
  pay = StockPrice + 1;
  market!buy(.pid);
}
atomic /* Buying processed */
{
  market?bought(eval(.pid)) ->

do /* Rating update */
:: trader < T ->
   Rating[trader] = Rating[trader] + Stocks[trader];
   trader++;
:: else -> trader=0; break;
od;

Stocks[id]++;
Money = Money - pay;
Appendix B. ProMeLa Models

\[ StockPrice = StockPrice + 2; \]

\* PriceTrend update */
\[ if \]
\[ :: temp == 0 -> PriceTrend = 0; \]
\[ :: temp == 1 -> PriceTrend = 0; \]
\[ :: temp == 2 -> PriceTrend = 1; \]
\[ :: temp == 3 -> PriceTrend = 1; \]
\[ :: temp == 4 -> PriceTrend = 2; \]
\[ :: temp == 5 -> PriceTrend = 2; \]
\[ fi; \]
\[ } :: atomic /* Attempt for selling */
\[ { \]
\[ (Stocks[id] > 0) && (PriceTrend == 0) ->
\[ obtain = StockPrice - 1; \]
\[ market!sell(_pid); \]
\[ } \]

atomic /* Selling processed */
\[ { \]
\[ market?sold(eval(_pid)) ->
\[ Stocks[id]--; \]
\[ Money = Money + obtain; \]
\[ StockPrice = StockPrice - 2; \]
\[ do /* Rating update */
\[ :: trader < T ->
\[ Rating[trader] = Rating[trader] - Stocks[trader]; \]
\[ trader++; \]
\[ :: else -> trader=0; break; \]
\[ od; \]

\* PriceTrend update */
\[ if \]
\[ :: temp == 0 -> PriceTrend = 0; \]
\[ :: temp == 1 -> PriceTrend = 0; \]
\[ :: temp == 2 -> PriceTrend = 1; \]
\[ :: temp == 3 -> PriceTrend = 1; \]
\[ :: temp == 4 -> PriceTrend = 2; \]
\[ :: temp == 5 -> PriceTrend = 2; \]
\[ fi; \]
\[ } \]

active[50] proctype S () /* Stocks */
Appendix B. ProMeLa Models

{ byte subj;

endBuy:
    do :: atomic /* Being bought */
    {
        market?buy(subj);
        market!bought(subj);
    }

endSell:
    atomic /* Being sold */
    {
        market?sell(eval(subj));
        market!sold(subj);
    }
    od

init
    { /* PT initialisation */
        PT[0]=0; PT[1]=0;

        run C (0);
        run C (1);
        run C (2);
        run C (3);
        run C (4);
        run C (5);

        run F (6,175);
        run F (7,191);
        run F (8,199);
        run F (9,201);
        run F (10,209);
        run F (11,225);

        go = 1;
    }

B.8 Inter-chartist variability added

mtype ( buy, bought, sell, sold );
chan market = [0] of { mtype, byte };

#define M 1000; /* Initial amount of speculative money */
short StockPrice = 150; /* Stock price and its initialisation */
byte PriceTrend[3] = 1; /* Price trend and its initialisation to a neutral value */
byte PT[5]; /* Auxiliary array for keeping the history of the price trend */
#define T 12; /* Number of traders */
byte Stocks[12]; /* Array for keeping the number of stocks held by each trader */
short Rating[12]; /* Array for keeping the rating of each trader */

bit go = 0;

proctype F (byte id; byte StockValue) /* Fundamentalists */
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;
    go == 1;
end:

do /* Attempt for buying */
:: atomic
{
    (StockPrice < StockValue) && (Money > StockPrice) ->
    pay = StockPrice + 1;
    market!buy(_pid);
}

atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->

    do /* Rating update */
:: trader < T ->
    Rating[trader] = Rating[trader] + Stocks[trader];
    trader++;
:: else -> trader=0; break;
od;

    Stocks[id]++;
    Money = Money - pay;
    StockPrice = StockPrice + 2;

    /* PriceTrend update */
    if
:: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
  }
Appendix B. ProMeLa Models

```proclaim
:: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
:: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
:: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;
}
:: atomic /* Attempt for selling */
{
(Stocks[id] > 0) && (StockPrice > StockValue) ->
obtain = StockPrice - 1;
market!sell(_pid);
}
atomic /* Selling processed */
{
market?sold(eval(_pid)) ->

Stocks[id]--;
Money = Money + obtain;
StockPrice = StockPrice - 2;
do /* Rating update */
:: trader < T ->
    Rating[trader] = Rating[trader] - Stocks[trader];
    trader++;
:: else -> trader=0; break;
od;
/* PriceTrend update */
if
:: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
:: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
:: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
:: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;
}
proctype C (byte id; byte strategy) /* Chartists */
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;
go == 1;
    end:
    do
```
Appendix B. ProMeLa Models

:: atomic /* Attempt for buying */
{
    (PriceTrend[strategy] == 2) && (Money > StockPrice) ->
    pay = StockPrice + 1;
    market!buy(_pid);
}

atomic /* Buying processed */
{
    market?bought(eval(_pid)) ->

do /* Rating update */
    :: trader < T ->
        Rating[trader] = Rating[trader] + Stocks[trader];
        trader++;
    :: else -> trader=0; break;
    od;

    Stocks[id]++;
    Money = Money - pay;
    StockPrice = StockPrice + 2;

    /* PriceTrend update */
    if
      :: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
      :: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
      :: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
      :: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
      :: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
      :: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
    fi;
}

:: atomic /* Attempt for selling */
{
    (Stocks[id] > 0) && (PriceTrend[strategy] == 0) ->
    obtain = StockPrice - 1;
    market!sell(_pid);
}

atomic /* Selling processed */
{
    market?sold(eval(_pid)) ->

    Stocks[id]--;
    Money = Money + obtain;
    StockPrice = StockPrice - 2;

do /* Rating update */
    :: trader < T ->
        Rating[trader] = Rating[trader] - Stocks[trader];
        trader++;
    :: else -> trader=0; break;
Appendix B. ProMeLa Models

od;

/* PriceTrend update */
if
:: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
:: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
:: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
:: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;
od

active[50] proctype S () /* Stocks */
{
  byte subj;
endBuy:
do
:: atomic /* Being bought */
{
  market?buy(subj);
  market!bought(subj);
};
endSell:

  atomic /* Being sold */
{
   market?sell(eval(subj));
   market!sold(subj);
};
} od

init
{
  /* PT initialisation */
  PT[0]=0; PT[1]=0;
run C (0,0);
run C (1,0);
run C (2,1);
run C (3,1);
APPENDIX B. ProMeLa Models

run C (4,2);
run C (5,2);
run F (6,175);
run F (7,191);
run F (8,199);
run F (9,201);
run F (10,209);
run F (11,225);
go = 1;
}

B.9 The final model (with step count)

mtype { buy, bought, sell, sold };
chan market = [0] of { mtype, byte };
#define M 1000; /* Initial amount of speculative money */
short StockPrice = 150; /* Stock price and its initialisation */
byte PriceTrend[3] = 1; /* Price trend and its initialisation to a neutral value */
byte PT[5]; /* Auxilary array for keeping the history of the price trend */
#define T 12; /* Number of traders */
byte Stocks[12]; /* Array for keeping the number of stocks held by each trader */
short Rating[12]; /* Array for keeping the rating of each trader */
byte Steps; /* Temporary variable counting price changes up to 201 steps */
bit go = 0;

proctype F (byte id; byte StockValue) /* Fundamentalists */
{
    short Money = M;
    short pay, obtain;
    byte trader, temp;
go == 1;
end:

    do :: atomic /* Attempt for buying */
    {
        (StockPrice < StockValue) && (Money > StockPrice) ->
        pay = StockPrice + 1;
        market!buy(_pid);
    }
    atomic /* Buying processed */
}
market?bought(eval(_pid)) ->

    do /* Rating update */
        :: trader < T ->
            Rating[trader] = Rating[trader] + Stocks[trader];
            trader++;
        :: else -> trader=0; break;
    od;

    Stocks[id]++;
    Money = Money - pay;
    StockPrice = StockPrice + 2;

    /* PriceTrend update */
    if
        :: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
        :: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
        :: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
        :: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
        :: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
        :: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
    fi;

    /* Steps count */
    if
        :: Steps <= 200 -> Steps++;
        :: else -> skip;
    fi;
}

:: atomic /* Attempt for selling */
{
    (Stocks[id] > 0) && (StockPrice > StockValue) ->
        obtain = StockPrice - 1;
        market!sell(_pid);
}

atomic /* Selling processed */
{
    market?sold(eval(_pid)) ->

        Stocks[id]--;
        Money = Money + obtain;
        StockPrice = StockPrice - 2;

    do /* Rating update */
        :: trader < T ->
            Rating[trader] = Rating[trader] - Stocks[trader];
            trader++;
        :: else -> trader=0; break;
    od;
Appendix B. ProMeLa Models

/* PriceTrend update */
if :: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
:: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
:: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
:: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
:: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;

/* Steps count */
if :: Steps <= 200 -> Steps++;
:: else -> skip;
fi;
}
}

proctype C (byte id; byte strategy) /* Chartists */
{
short Money = M;
short pay, obtain;
byte trader, temp;
go == 1;
end:
do :: atomic /* Attempt for buying */
{
(PriceTrend[strategy] == 2) && (Money > StockPrice) ->
pay = StockPrice + 1;
market!buy(_pid);
}
atomic /* Buying processed */
{
market?bought(eval(_pid)) ->
do /* Rating update */
:: trader < T ->
Rating[trader] = Rating[trader] + Stocks[trader];
trader++;
:: else -> trader=0; break;
od;
Stocks[id]++;
Money = Money - pay;
StockPrice = StockPrice + 2;
}
Appendix B. ProMeLa Models

/* PriceTrend update */
if
  :: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
  :: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
  :: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
  :: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
  :: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
  :: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;

/* Steps count */
if
  :: Steps <= 200 -> Steps++;
  :: else -> skip;
fi;

*: atomic /* Attempt for selling */
{
  (Stocks[id] > 0) && (PriceTrend[strategy] == 0) ->
  obtain = StockPrice - 1;
  market!sell(_pid);
}
atomic /* Selling processed */
{
  market?sold(eval(_pid)) ->
  Stocks[id]--;
  Money = Money + obtain;
  StockPrice = StockPrice - 2;

  do /* Rating update */
    :: trader < T ->
      Rating[trader] = Rating[trader] - Stocks[trader];
      trader++;
    :: else -> trader=0; break;
  od;

  /* PriceTrend update */
if
  :: temp == 0 -> PriceTrend[0]=0; PriceTrend[1]=0; PriceTrend[2]=0;
  :: temp == 1 -> PriceTrend[0]=1; PriceTrend[1]=0; PriceTrend[2]=0;
  :: temp == 2 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=0;
  :: temp == 3 -> PriceTrend[0]=1; PriceTrend[1]=1; PriceTrend[2]=2;
  :: temp == 4 -> PriceTrend[0]=1; PriceTrend[1]=2; PriceTrend[2]=2;
  :: temp == 5 -> PriceTrend[0]=2; PriceTrend[1]=2; PriceTrend[2]=2;
fi;
APPENDIX B. ProMeLa Models

/* Steps count */
if :: Steps <= 200 -> Steps++;
:: else -> skip;
fi;
}
}
}

active[50] proctype S () /* Stocks */
{
byte subj;
endBuy:

do :: atomic /* Being bought */
{
market?buy(subj);
market!bought(subj);
};
endSell:

atomic /* Being sold */
{
market?sell(eval(subj));
market!sold(subj);
};
od
}

init
{
/* PT initialisation */
PT[0]=0; PT[1]=0;
run C (0,0);
run C (1,0);
run C (2,1);
run C (3,1);
run C (4,2);
run C (5,2);
run F (6,175);
run F (7,191);
run F (8,199);
run F (9,201);
run F (10,209);

160
run F (11,225);

/* run C (0,0);
 run C (1,0);
 run C (2,0);
 run C (3,0);
 run C (4,1);
 run C (5,1);
 run C (6,1);
 run C (7,1);
 run C (8,2);
 run C (9,2);
 run C (10,2);
 run C (11,2);

 run F (12,170);
 run F (13,184);
 run F (14,192);
 run F (15,196);
 run F (16,198);
 run F (17,199);
 run F (18,200);
 run F (19,201);
 run F (20,204);
 run F (21,208);
 run F (22,216);
 run F (23,230); */

go = 1;
}
Appendix C

Content of the CD

The attached CD contains the PDF file of this thesis, named Zimmerova-DP.pdf, and a number of folders with ProMeLa models and verification results discussed in Chapter 4 of this thesis. In total, there are ten folders containing the evolving models, and about 100 files with verification reports. The structure of the folders follows the structure of Chapter 4, and is the following.

01-simple-traders
02-speculative-money
03-two-types-of-traders
04-strategies-evaluation
05-negotiation-time
06-init-speculative-money
07-stockvalue-changes
08-fundamentalists-variability
09-chartists-variability
10-different-proportions
APPENDIX C. CONTENT OF THE CD