Bottom-up design of strategic options as finite automata.

OLIVEIRA, F.S.

2010

Bottom-up Design of Strategic Options as Finite Automata

Fernando S. Oliveira

Warwick Business School, University of Warwick, UK and Cempre, Faculdade de Economia, Universidade do Porto, Portugal; Gibbet Hill Road, Coventry, CV4 7AL, UK

Tel: +44(0)2476 522709, fax: +44(0)2476 524539, Email: Fernando.Oliveira@wbs.ac.uk.

Abstract

In this paper we look at the problem of strategic decision making. We start by presenting a new formalisation of strategic options as finite automata. Then, we show that these finite automata can be used to develop complex models of interacting options, such as option combinations and product options. Finally, we analyse real option games, presenting an algorithm to generate option games (based on automata).

Keywords: automata, game, real options, strategy.

1. Introduction

The theory of investment is, in its majority, a one decision-maker problem, however, a new method of modelling strategic options has started to be accepted, the real options approach. In this line of research, Luerhrman (1998) presented the concept of strategy as a bundle of real options and McGarth (1999) argued that the entrepreneurial activity can be characterised as real options in which companies are bundles of real options. More recently the use of real options in a game setting has proved invaluable to model strategic decision making, e.g., Smit and Trigeorgis (2004). In this paper, we develop a new method to formalize the strategy problem that enables a formal treatment of complex, non-stationary interactions between different decision-makers.

One of the contributions of our research is the formalisation of the strategic planning problem incorporating bounded rationality, and behavioural issues, in strategic games (Section 4). This allows the evaluation of strategies taking into account the long-term effects of the opponents' behaviour.

What is the advantage of formalising the strategy problem using finite automata?

The traditional way to solve the game, i.e. to calculate how each player will play (Smit and Ankum, 1993), is to construct the decision tree and by backward induction compute the set of equilibrium strategies. Note that a decision tree is just a special case of a finite automaton in which there are no loops (it is a direct acyclic graph). In this case each player is assumed to be perfectly rational (there is no need for a player to build a model of his opponent's behaviour) and the decisions are sequential. Within this traditional game theoretical approach the agents are given too much information: they know how the *nature* behaves and they also know the outcome of each pair of decisions (one from each player). This is an extremely strong assumption on the players' rationality – they can compute the output of the interactions between both players' decisions along the tree. Another assumption underlying the traditional approach is that the players always choose rationally, each one always maximises his payoff.

The finite automata, on the other hand, are much less restrictive on the rationality assumptions underlying human and organisational behaviour: a certain automaton is a rule (more or less complex) of behaviour that does not assume any optimal reasoning or knowledge about the other player's behaviour.

The main contribution of this paper is to develop algorithms to design complex real options (Section 5) such as product options and option combinations, and to design real-option games (Section 6). These algorithms incorporate bounded rationality by modelling a firm as an entity resulting from the interactions between the rules of behaviour underlying the different decisions and projects considered by the firm. In Section 5 we show how to arrange several options and the Net Present Value rule into the bundle of options that describes a company. We prove the existence of such bundle of options and present an algorithm for their development, illustrating this algorithm with an example. In Section 6 we present an algorithm for the development of real option games, analysing an illustrative example and proving the existence of equilibrium in this game.

The paper proceeds by analysing the real options approach, and the way it has been used to study strategic decision making (in Sections 2 and 3), and then we proceed by presenting how to model strategic decisions using finite automata (in Section 4). In Section 5 we analyse real option design from simple options, and in Section 6 we show how to use automata theory to develop real option games. Section 7 concludes the paper.

2. The Real Options Approach to Strategic Decision Making

The traditional way of evaluating investment projects, the discounted cash flows (*DCFs*), studies each project taking only into account the individual cash flows (only if the net present value of the expected cash flows is positive the project should be undertaken). MacDonald and Siegle (1986) in analysing the optimal time of investment in an irreversible project have shown that the correct calculation of the timing to invest in a project results from the comparison of the value of investing today with the value of investing at all possible times in the future, and they derived the formulas to calculate the value of the option to invest, and to calculate the optimal time to invest.

Furthermore, Pindyck (1991) identified two main problems in the *DCFs* approach: firstly, the existence of sunk costs that cannot be recovered; secondly, the possibility of delaying an investment that gives the firm an opportunity to wait for new information before taking the final decision of commit to the project. For these reasons, Pinyck defended that the presence of uncertainty, irreversibility, and of a delay option, makes the *DCFs* method inadequate to evaluate a new project.

Dixit (1992) also argued against the *DCFs* theory of investment, since in reality firms invest in projects with a return rate three to four times the cost of capital. Firms only invest if the price rises well above the long run average cost (on the other hand firms with operating losses stay in business for a long time): i.e., the option of waiting has a positive value. Dixit (applying the option price theory to evaluate investment projects) concludes that the value of waiting may significantly increase the rate of return demanded in an investment, and concludes that the real options theory explains why the rates of return demanded by companies are much higher than it would be predicted by the *DCFs* approach.

Overall, following Trigeorgis (1998, pp. 2), the literature has developed the following main types of real options: option to defer, management holds an option to buy valuable land or resources; time-to-build option, the investment is divided into different stages creating the option to abandon the project if the new information is unfavourable to future developments; option to alter the operating scale, the options to expand, contract, shut down or restart; option to abandon, if the market conditions get worse than expected management has the option to abandon definitely the project, receiving the resale value of the assets owned; option to switch, in which the agent has the option to change the input (or output) mix if the prices or demand change; growth options, in this case an early investment may be seen as the start of a chain of interrelated projects given access to future growth opportunities; multiple interacting options.

Investment projects usually present some of the types of options presented above interacting between them.

Some more recent research on real-options reflects the need to adapt the basic theoretical framework to solve real problems. For example:

- Cortazar, Schwartz and Salinas (1998) presented an application of real-options to the valuation of environmental investments. Their analysis showed that firms, in industries with high output price volatility, would be more willing to operate at low output levels (reducing the emissions) than to invest in environmental protection technologies.

- Bollen (1999) developed an option valuation framework that explicitly incorporates a product life cycle (using a regime switching process) assuming a monopolist firm.

- Huchzermeier and Loch (2001) applied real-options theory to evaluate flexibility in R&D projects. They identify five types of R&D uncertainty (market payoffs, project budgets, product performance, market requirements and project schedules). They have shown that operational uncertainty (in product performance, market requirements and schedule adherence) may reduce the real option value.

- Valuation (e.g., Thompson et al., 2004) and optimal operation (e.g., Tseng, and Barz, 2002) of electricity plants.

- Analysis of leasing problems (e.g., Kenyon and Tompaidis, 2001).

- Study of operational risk management in the mining industry (e.g., Kardia and Ernst, 2001).

Kulatilaka and Perotti (1998) formulated a very important critique to the real options approach: it assumes that the product market is perfectly competitive, and that the firm has a monopoly over the investment opportunity (for an exhaustive critique of real-options see Lander and Pinches, 1998).

The Kulatilaka and Perotti's critique is a very important one because in order to model strategic options we need to take into account the interactions between the different decision makers. This issue has been approached in several important papers, such as:

- Smit and Ankum (1993) modelled corporate investment in a duopolistic industry showing that under competition there is a lower tendency to postpone projects. They have also shown that under a duopoly the timing strategy of the incumbent is coupled with the tactics of its rival. Smit and Ankum assume that both firms have the same tactics and are perfectly rational (assuming also complete information).

- Grenadier (1996) also modelled the interaction between players in a duopoly (in the real estate industry) showing, under the assumption of perfect rationality, that it may be an optimal strategy to have periods of overbuilding with declining demand.

- Kulatilaka and Perotti (1998) modelled the interaction between the two companies as a one shot first-entry game (a model of strategic growth options in a duopoly) strongly emphasizing the value of the initial investment as the acquisition of opportunities relative to competitors. They show that under imperfect competition the effect of uncertainty on the value of a growth option is ambiguous, since oligopolistic firms react to better market conditions increasing both quantities and prices.

Next, in section 3 we present the theory of finite automata, before applying it to modelling real options.

3. Finite Automata Games

In this paper we develop the use of strategic options in games. A first development of the theory is to consider the existence of companies boundedly rational: given the limits of computation and access to information each firm does not know the behaviour of his opponents and has to define rules of behaviour that allows the firm to adapt to his opponents choices and possible developments of the game.

Rubinstein (1986) used finite automata as a method to model bounded rationality. The repeated game modelled with finite automata attempts to capture the bounded rationality of the players involved in the game, analysing automata with a bounded number of states.

A finite automaton is a decision rule, or a strategy, consisting of a finite set of states, a transition function (which defines the rules of transition between states) and a behavioral function (defining how the player behaves in each state of the automaton), see Rubinstein (1986) and Hopcroft and Ullman (1979).

In automata games we can also compute the equilibrium. This issue has been addressed within game theory in several different ways, one of which is the automata game (Rubinstein, 1986). This line of research has analyzed the complexity of computing the best response automaton (Gilboa, 1988; Banks and Sundaram, 1990; Piccione, 1992) and the issue of the equilibrium and coordination in automata games (Abreu and Rubinstein, 1988; Gilboa and Zemel, 1989; Newman, 1998; Eliaz, 2003; Gossner and Hernandez, 2003).

In this section, we start by formalizing an *N*-player automata game (Definition 3.1), which defines the structure that regulates how players interact. We then formalize the definition of automaton.

Definition 3.1: *An automata game is a 5-tuple* $G = \left(\mathrm{N},\ \{\mathrm{Z}^{\mathrm{i}}\}_{i=1}^{N},\ \{u^i\}_{i=1}^{N}, \{Q^i\}_{i=1}^{N}, \{\Sigma^i\}_{i=1}^{N} \right)$ *in which N denotes the number of players.* Σ^i *is a non-empty set of possible actions of player i.* Z^i *represents a finite non-empty set of possible outcomes of the game in which each* $z^i \in Z^i$ *is dependent on the actions of each player,* $z^i = z(a^1,...,a^i,...,a^N)$, *and* $a^i \in \Sigma^i$ *represents the player i's actions and for all* $j \neq i$ *the action* $a^j \in \sum^j$ *represents his opponents' actions.* $u^i = u(z^i)$ *represents the utility function of player i, i.e., it is the payoff a player i receives from his action.* Q^i *stands for a finite non-empty set of internal states of player i.*

From Definition 3.1, it follows that the outcomes of the game represent the information received by each player at the end of every stage. This information, or outcome, is a function of the actions of each player in the stage game, and it is different for each one of the players, as each one only knows the outcome of his own actions. The behavior of each player in any one of his internal states is defined by the automaton presented in Definition 3.2.

Definition 3.2: *A finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *is a 6-tuple in which* q_0^i *is the initial internal state,* Σ^i *is the set of all the possible actions,* Z^i *represents the set of possible outcomes,* δ^i *is a transition function* $\left(\delta^i : Q^i \times Z^i \to Q^i\right)$ *and* λ^i *is a behavioral function* $\left(\lambda^i : Q^i \to \Sigma^i\right)$ *associating one action with each possible internal state.*

The interaction between the different automata involved in the game can be described as follows. At stage 1 each player *i* plays $\lambda^i\left(q_0^i\right)$. At a stage $t \geq 1$, after receiving an outcome $z^i = Z\left(a^1, ..., a^i, ..., a^N\right)$, the state of automaton A^i changes from the state q_t^i to the state $\delta^i\left(q_t^i, z_t^i\right)$. Then, each player *i* chooses a new move, $\lambda^i\left(q_{t+1}^i\right)$.

The repeated game with finite automata tries to capture the bounded rationality of the players involved in the game by considering automata with a bounded number of states. Papadimitriou (1992) argued that the limitations on the number of states do *not* capture the bounded rationality: bounded rationality is a restriction on the capabilities of the agent to *design* a strategy, while the bound on the number of states only captures a limitation on the *implementation* of a strategy. This gives rise to two different types of complexity: a design complexity and an implementation complexity. There is a trade-off between the two types of complexity.

Abreu and Rubinstein (1988) suggested that one possible application of the finite automata models is to modelling the *organization of bureaucracies* in which the managers seek simple rules that can be implemented mechanically by each employee. The automaton is viewed as the set of managerial instructions.

Next, in Section 4, we show how real options can be modelled as automata.

4. Modelling Real Options with Finite Automata

In this section, we show how finite automata can be used to model real options. Although apparently trivial, the model of real options as finite automata shows that this theory can be extended and replicated using finite automata.

In our figures we use the symbols presented in Figure 4.1.

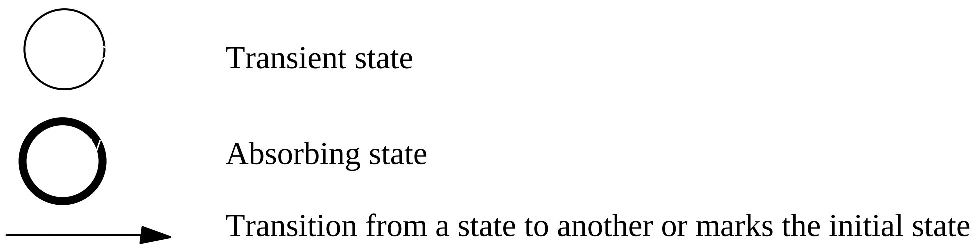
Transient state
Absorbing state
Transition from a state to another or marks the initial state

Figure 4.1: Symbols used to draw automata

Definition 4.1: An automaton's state is said transient if it contains at least one transition to any other state.

Definition 4.2: An automaton's state is said absorbing if it contains no transition to any other state.

Let us start by analysing *nature*. As in current options literature we define *nature* has a random sequence of events, and most specifically, by following Dixit and Pindyck (1994, p. 68), as a geometric Brownian motion, represented in Figure 4.2 and Definition 4.3. This is just an example of a possible finite automaton that replicates stochastic moves. Different stochastic automata may be a better choice for different types of problems.

Definition 4.3: *The geometric Brownian motion can be represented as a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that:* $Q^i = \left\{S_0, S_u, .S_d\right\}$; $q_0^i = S_0$; $\Sigma^i = \left\{p, 1-p\right\}$ *is the set of all the possible action, in this case represented by the probabilities of moving up (p) or down (1-p) and* $Z^i = \left\{u, d\right\}$ *represents the set of possible outcomes from moving up or down, in which* $d = \frac{1}{u}$ *and u>1. Further,* $\delta^i\left(S_{t-1}, u\right) = S_u$ *and* $\delta^i\left(S_{t-1}, d\right) = S_d$ *represent the transition function, such that* $S_u = uS_{t-1}$ *and* $S_d = dS_{t-1}$*. Finally,* $\lambda^i \in \Sigma^i$.

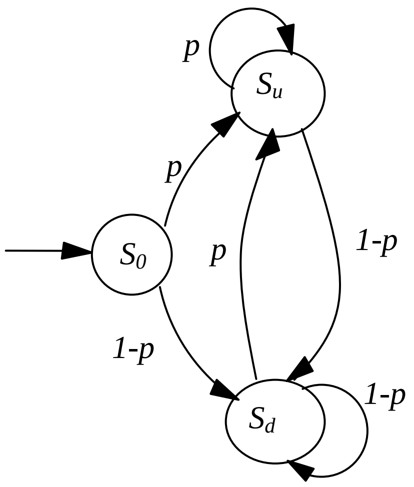

Figure 4.2: Nature's automaton for the Geometric Brownian Motion

Next, we formalise as an automaton the decision rule of investment based on the Net Present Value of a series of discounted cash flows. This traditional investment strategy postulates that an investor, when planning his investments, would evaluate his cash flows and, if they were non-negative, he would invest (I), otherwise he would abandon the project (NI). This decision rule can be modelled using the automaton in Figure. 4.3. We formalize this concept in Definition 4.4.

Definition 4.4: *The NPV rule is a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that:* $Q^i = \left\{0, I, NI\right\}$; $q_0^i = 0$; $\Sigma^i = \left\{I, NI\right\}$ *is the set of all the possible actions and* $Z^i = \left\{NPV \geq 0, NPV < 0\right\}$ *represents the set of possible outcomes. Further,*

$\delta^i(0, NPV \geq 0) = I$, $\delta^i(0, NPV < 0) = NI$, $\delta^i(I,.) = I$ *and* $\delta^i(NI,.) = NI$ *represent the transition function. Finally,* λ^i *is a behavioral function such that for any state in* $D_j \in Q^i$ $\lambda^i(D_j) \in \{I, NI\}$.

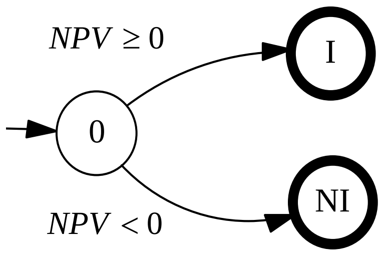

Figure 4.3: The Automaton Describing the NPV Rule

We now formulate real options as finite automata. The first option to be represented as an automaton is the *option to defer*. A very first approach to model this option is to look at the basic decision, which is based on the comparison of the net present value (NPV) at a time *t* with the NPV of the same investment at a time *t+k*, in which *k* represents a future time. If the present value of the investment started at *t+k* is higher than the present value of the investment started today the investment is delayed one period. At the period *t+1* a new evaluation of the *NPV* will take place. This automaton is represented in Figure 4.4 and formalised in Definition 4.5.

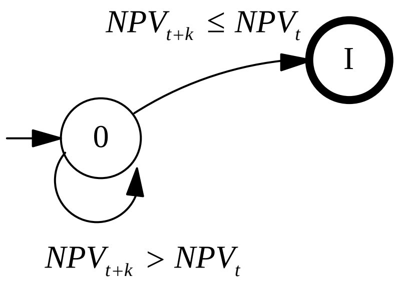

Figure 4.4: Automaton Describing the Option to Defer

Definition 4.5: *The option to Defer an investment is a finite automaton* $A^i = (Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i)$ *such that:* $Q^i = \{0, I\}$*;* $q_0^i = 0$*;* $\Sigma^i = \{I, NI\}$ *is the set of all the* *possible actions and* $Z^i = \{NPV_{t+k} > NPV_t, NPV_{t+k} \leq NPV_t\}$ *represents the set of possible outcomes. Furthermore, the following rules represent the transition function:* $\delta^i(0, NPV_{t+k} \leq NPV_t) = I$, $\delta^i(0, NPV_{t+k} > NPV_t) = NI$ *and* $\delta^i(I,.) = I$ *. Finally,* λ^i *is a behavioral function such that for any state in* $D_j \in Q^i$ *we have* $\lambda^i(D_j) \in \{I, NI\}$.

Moreover, the *option to abandon* has the same structure of the option to defer. The only difference are the instruments, instead of an irreversible investment the company has the possibility of irreversibly abandon (A) the project, see Figure 4.5.

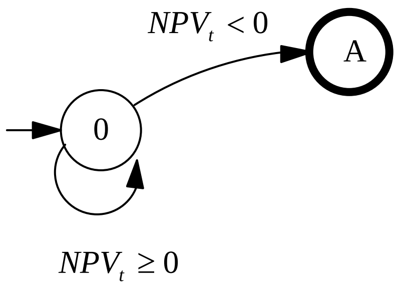

Figure 4.5: Automaton for the Option to Abandon

Another important real option is the *option to alter the operating scale*, assuming that the investment is reversible, that the firm is already operating, and that to abandon is not an option. The firm needs to decide if it should increase or decrease its operation scale, given its expectation for the business evolution.

As an example, let us analyse the option of a firm *F* producing a product *P* with an equipment *K* and labour *L*, with starting conditions are *P₀*, *K₀* and *L₀*. The firm has to decide if it wants to increase or decrease its operations scale given its expectations *E*, regarding the evolution of the market, see equation 4.1.

$$\begin{cases} \text{Increase scale } (\Delta^+ K, \Delta^+ L, \Delta^+ P) & \Leftarrow E > 0 \\ \text{Constant scale } (\Delta^0 K, \Delta^0 L, \Delta^0 P) & \Leftarrow E = 0 \\ \text{Decrease scale } (\Delta^- K, \Delta^- L, \Delta^- P) & \Leftarrow E < 0 \end{cases} \tag{4.1}$$

Let *0*, *Inc*, and *Dcr* represent, respectively, Do Nothing, Increase scale, and Decrease scale. The automaton for alter the operation scale can be represented as shown in Figure 4.6.

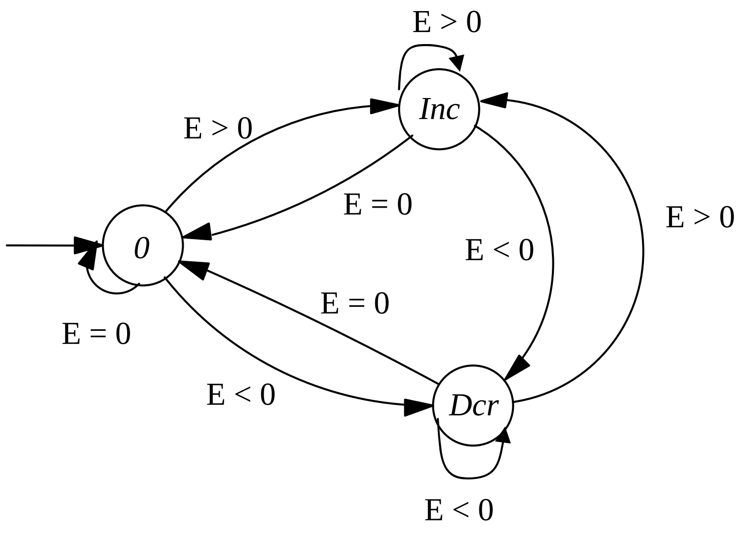

Figure 4.6: Automaton for the option to Alter Scale

In this automaton there is no absorbing state: the process of rescaling will be continuously reshaping the company in reply to management expectations. There is no definition of the final scale of the firm only the rates of change are fixed. Definition 4.6 formalises the option to alter scale. It is noteworthy that Definition 4.6 covers a number of possible actions, expectations and internal structures of the automaton which are much more complex than the ones presented in Figure 4.6.

Definition 4.6*: The option to Alter the Scale of a project is a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that:* $Q^i = \{D_1, D_2, ..., D_n\}$*;* $q_0^i = D_1$*;* $\Sigma^i = \{\Sigma_1, \Sigma_2, ..., \Sigma_m\}$ *is the set of all the possible actions for changing the scale of the firm, and in which* $Z^i = \{E_1, E_2, ..., E_k\}$ *represents the set of all possible expectations regarding the evolution of the business size. Further, for any j, r and s,* $\delta^i\left(D_j, \Sigma_r\right) = D_s$*, represents the transition function. Finally, and* λ^i *is a behavioral function such that for any state* D_j*:* $\lambda^i\left(D_j\right) \in \{\Sigma_1, \Sigma_2, ..., \Sigma_m\}$.

Firms also have the option to switch, i.e., to change the mixed of resources used maintaining the same scale of production. As an example, consider a firm F using the resources labour (*L*) and capital (*K*). The decision variable is the ratio capital on labour $\left(\frac{K}{L}\right)$ and it is a function of the ratio of the price of capital (K_p) on the price of labour (L_p), i.e., $\left(\frac{K_p}{L_p}\right)$.

In this case, assume that the firm has to decide which technologic combination to adopt of a finite vector of available *technologies*, $\{T_1,\ldots, T_T\}$. The value of the technologies and conversion costs (from one technology to another) will change depending on *nature's* moves. The agent will have some expectations on the technologies value *Switching* $\{E_{11}, E_{12}, ..., E_{21}, E_{22}, \ldots, E_{T1}, E_{T2}, ..., E_{TT}\}$, in which $E_{ii}=0$ and E_{ij} represents the expect profit from switching from technology *i* to technology *j*.

We represent a simple example of the finite automaton for the option to Switch agent's behaviour in Figure 4.7. Once again, a simple finite automaton is used to model a situation with no absorption state. In this case we assume three technologic combinations T_1, T_2, and T_3, moreover, in a given state T_i the firm is modifying its internal structure in order to implement the technology T_i.

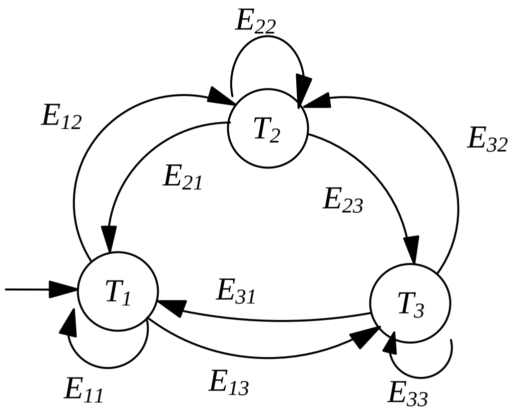

Figure 4.7: Automaton for the Option to Switch

Definition 4.7 represents this automaton using a more formal approach.

Definition 4.7*: Let the vector* {*T_1,…, T_T*} *represent the different combination of technologies and E_{kj} represents the expect profit from switching from technology k to technology j. The option to Switch is a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that* $Q^i = \{T_1, T_2, ..., T_T\}$*;* $q_0^i = T_1$*;* $\Sigma^i = \{T_1, T_2, ..., T_T\}$ *is the set of all the possible actions for implementing a given technological combination, and in which* $Z^i = \{E_{11}, E_{12}, ..., E_{TT}\}$ *represents the set of all possible expectations regarding value of switching technologies. Further, for any j, r and s,* $\delta^i\left(T_j, E_{jr}\right) = T_r$ *, represents the transition function. Finally,* λ^i *is a behavioral function such that for any state T_j:* $\lambda^i\left(T_j\right) \in \{T_1, T_2, ..., T_T\}$.

Next, we extend this methodology to the design of complex real options.

5. Designing Real Options

It is possible for an agent to develop an automaton that results from the *combination* of the basic automata presented in Section 4, such as the NPV rule (Definition 4.4), the option to defer (Definition 4.5) and option to abandon (Figure 4.5), the option to alter the scale (Definition 4.6), the option to switch (Definition 4.7). As these are rules of behavior they model bounded rational behavior, as they do not need to be the result of full optimization and, most likely, are the result of the firm's culture which evolved over time. The combination of automata is formalized in Definition 5.1.

Definition 5.1: *Let* $\Lambda^i = \left\{A_1^i, ..., A_N^i\right\}$ *represent the set of N basic automata available to player i. An* ***option combination*** *is a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that each element of this 6-tuple results from the agglutination of the elements of each basic automaton in the following way:*

a) $q_0^i \in \bigcup_{j=1}^{N} \left\{q_{0,j}^i\right\}$. *b)* $Q^i = \bigcup_{j=1}^{N} Q_j^i$. c) $\Sigma^i = \bigcup_{j=1}^{N} \Sigma_j^i$. d) $Z^i = \bigcup_{j=1}^{N} Z_j^i$.

e) *For every element $q^i_{j1}, q^i_{j2} \in Q^i$ and $z^i \in Z^i$ the transition function is well defined, i.e., for all j1 and j2 and z^i there is always a $\delta^i\left(q^i_{j1}, z^i\right) = q^i_{j2}$ if q^i_{j1} is transient or a $\delta^i\left(q^i_{j1}, z^i\right) = q^i_{j1}$ if q^i_{j1} is an absorbing state.*

Firstly, the states, actions and outputs used in each automaton depend on the strategies of each player. In any option combination the initial state is one of the initial states in the basic automata. In a given state the player can only choose actions possible in any of the automata to which that state belongs to. Absorbing (transient) states in the basic automata are absorbing (transient) states in the automata combination. Proposition 5.1 shows how a player may combine these different automata in order to develop his strategic plan and how this new rule of behavior is still an automaton, and therefore a representation of the rules governing a firm.

Proposition 5.1: *For the option combination, as defined in Definition 5.1, to be a well defined finite automaton $A^i = \left(Q^i, q^i_0, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ the following conditions are required to be met*:

a) For any ordered pair of states and outcomes $\left(q^i_1, z^i_2\right)$ in an option combination the transition function, $\delta^i\left(q^i_1, z^i_2\right)$, is such that if $q^i_1 \in Q^i_j$ then $z^i_2 \in Z^i_j$.

b) For all $q^i_{j1} \in Q^i$ and $\theta^i_{j1} \in \Sigma^i$, such that $q^i_{j1} \in Q^i_{j1}$, if $\theta^i_{j1} = \lambda^i\left(q^i_{j1}\right)$ then $\theta^i_{j1} \in \Sigma^i_{j1}$.

c) Every transition from any state in a given option must be mutually excluding or lead to the same state. For every state q^i, and outcomes z^i_1, z^i_2, if for at least one instance of z^i_1, z^i_2 they are both true, i.e., $z^i_1 \cap z^i_2$ is true, then $\delta^i\left(q^i, z^i_1\right) = \delta^i\left(q^i, z^i_2\right)$.

Proof: *a*) We prove this step by contra-positive. Assume that for any ordered pair of states and outcomes $\left(q^i_1, z^i_2\right)$ in a option combination, the transition function, $\delta^i\left(q^i_1, z^i_2\right)$, is such that $q^i_1 \in Q^i_{j1}$, $q^i_1 \notin Q^i_{j2}$, $z^i_2 \notin Z^i_{j1}$ and $z^i_2 \in Z^i_{j2}$. Consequently, in this case, $\delta^i_{j1}\left(q^i_1, z^i_2\right)$ is not in automaton A^i_{j1}, and $\delta^i_{j2}\left(q^i_1, z^i_2\right)$ is not defined in automaton A^i_{j2}, and therefore, $\delta^i\left(q^i_1, z^i_2\right)$ is not defined in the combination of the basic automata. Hence, for every well define transition in the combination automata any pair (state, outcome) belongs at least to one basic automaton.

b) We need to prove that this is a condition for the behavior function to be well defined. Again, this proposition is proved by contra-positive. Assume that for all $q^i_{j1} \in Q^i$ such that $q^i_{j1} \in Q^i_{j1}$, and for all $\theta^i_{j1} \in \Sigma^i$, $\theta^i_{j1} = \lambda^i\left(q^i_{j1}\right)$ and that $\theta^i_{j1} \notin \Sigma^i_{j1}$. Consequently, $\theta^i_{j1} \notin \Sigma^i_{j1}$ then $\theta^i_{j1} = \lambda^i_{j1}\left(q^i_{j1}\right)$ is nor defined in the basic automaton A^i_{j1} and therefore the basic automaton is not well defined. Concluding, condition *b*) is necessary for the option combination to be a well defined automaton.

c) Again, this proposition is proved by contra-positive. Assume that for every state q^i, and outcomes z^i_1, z^i_2, at least one instance of z^i_1, z^i_2 they are both true, and that $\delta^i\left(q^i, z^i_1\right) \neq \delta^i\left(q^i, z^i_2\right)$. Consequently, there is an instance of z^i_1, z^i_2 for which in state q^i there are at least two simultaneous transitions, which contradicts the definition transition function which only has a possible transition at any given time. Q.E.D.

Moreover, a necessary condition for the existence of an option combination is that some of the states in the basic automata are the same. This is formalised in Proposition 5.2.

Proposition 5.2: *Let $\Lambda^i = \left\{A^i_1, ..., A^i_N\right\}$ represent the set of N basic automata available to player i. A necessary condition for the existence of an* ***option combination*** $A^i = \left(Q^i, q^i_0, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *is that there is at least one state q^i that is a member of at two different set of states, Q^i_{j1} and Q^i_{j2}, belonging to the basic automata .*

Proof: The proof is by contra-positive. Assume that there are no common states among the several different automata, i.e., assume that for any pair of set of states Q^i_{j1} and Q^i_{j2} belonging to the basic automata, for all q^i_{j1} and q^i_{j2}: $q^i_{j1} \in Q^i_{j1}$, $q^i_{j2} \in Q^i_{j2}$, $q^i_{j2} \notin Q^i_{j1}$ and $q^i_{j1} \notin Q^i_{j2}$.

Moreover, from Proposition 5.1.g) we know that *for all* $q^i_{j1} \in Q^i$ *such that* $q^i_{j1} \in Q^i_{j1}$ *and* $\theta^i_{j1} \in \Sigma^i$, *if* $\theta^i_{j1} = \lambda^i\left(q^i_{j1}\right)$ *then* $\theta^i_{j1} \in \Sigma^i_{j1}$. *Therefore, together with Proposition 5.1.f)* $\delta^i\left(q^i_1, z^i_2\right)$, *is such that if* $q^i_1 \in Q^i_j$ *then* $z^i_2 \in Z^i_j$, *it follows that there is no state* $q^i_2 = \delta^i\left(q^i_1, z^i_2\right)$ *such that* $q^i_{j2} \notin Q^i_{j1}$, as all the transitions occur within the same automaton. Q.E.D.

Proposition 5.3: *The option to switch cannot be combined with the options to defer, the NPV rule, the option to abandon, or the option to alter scale.*

Proof: From Proposition 5.2 and Definitions 4.2-4.5 it follows that as the option to switch does not have any common state with any other option, then no combination is possible. Q.E.D.

In practical terms this result implies that the option to switch relates to a different stage of decisions, and possibly a different type of problem, than the ones modelled by the options to defer, abandon, the NPV rule or the option to alter scale: whilst the options to abandon, defer and the NPV rule relate to a decision on entering or abandon a project, the option to switch relates to the current management of a running project. Similarly, the options to switch and to alter scale are not related to each other: the option to switch works at an operational level whereas the option to alter the scale is applied at a strategic level.

Let us look at a simple example of automaton combination by combining the automata in Figures 4.3-4.6, see Figure 5.1. It should be noticed that Proposition 5.1.h) rules out any combination between the option to alter scale and any of the following, NPV rule, Option to Abandon or Option to Defer, see Proposition 5.4. The economic intuition is simple: you can alter the scale of a current project and simultaneously start or abandon a different project, but not the scale of the project in which you have not yet invested. This means that the actions available in the option to alter the scale cannot be used to extend the options to Defer, Abandon or to the NPV rule.

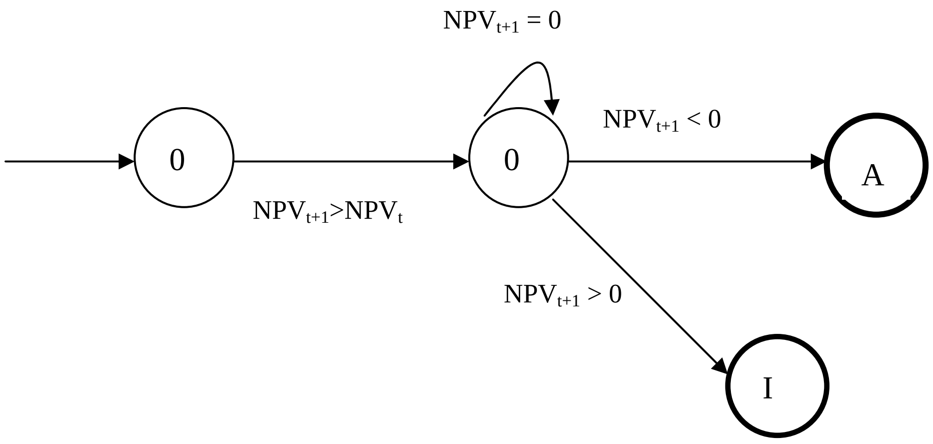

Figure 5.1: An Example Combining the Options to Defer, to Abandon and the NPV Rule

Proposition 5.4: *The option to Alter Scale cannot be combined with the options to Defer, the NPV rule and the option to Abandon.*

Proof: From Definitions 4.2-4.5 it follows the option to Alter Scale may have a common state $D_j = 0$ with the other options. For this common state, $D_j = 0$, there are several actions in the options to Alter Scale which do not exclude the actions in the options to Defer, Abandon, or in the NPV rule, {0, NI, I}. Consequently, by Proposition 5.1.h) this combination is not possible. Q.E.D.

Next, in Table 5.1, and as a result of Propositions 5.1-5.4 we present an algorithm to build option combinations.

We can now analyze the construction of product automata, see Definition 5.2 and Proposition 5.5.

Definition 5.2: *Let* $\Lambda^i = \{A_1^i, ..., A_N^i\}$ *represent the set of N basic automata available to player i, and let* $\prod$ *stand for the Cartesian product. An* ***option product*** *is a finite automaton* $A^i = (Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i)$ *such that each element of this 6-tuple results from the agglutination of the elements of each basic automaton in the following way:*

a) $q_0^i = \prod_{j=1}^{N} q_{0,j}^i$. *b)* $Q^i = \prod_{j=1}^{N} Q_j^i$. c) $\Sigma^i = \prod_{j=1}^{N} \Sigma_j^i$. d) $Z^i = \prod_{j=1}^{N} Z_j^i$.

e) $\delta^i : Q^i \times Z^i \to Q^i$. *For every element* $q_{j1}^i, q_{j2}^i \in Q^i$ *and* $z^i \in Z^i$ *the transition function is well defined, i.e., for all j1 and j2 and* z^i *there is always a* $\delta^i\left(q_{j1}^i, z^i\right) = q_{j2}^i$ *if* q_{j1}^i *is transient or a* $\delta^i\left(q_{j1}^i, z^i\right) = q_{j1}^i$ *if* q_{j1}^i *is an absorbing state.*

Table 5.1. The Option Combination Algorithm

Parameters:

B: Number of basic automata

A^k: The basic automaton *k,* $A^k = \left(Q^k, q_0^k, \Sigma^k, Z^k, \delta^k, \lambda^k\right)$

Compatible outcomes: $Compatible\left(Z_w, Z_y\right)$: $Z_w \cap Z_y$ is ***not*** an impossible event

Algorithm for option combination:

```
Let $C_j$ stand for the combined automaton at time $j$
    Step 1. For the current automaton $A^k$ and $C_j$ find all the states that have
        the same actions, ($q_a$, $q_c$), respectively.

    Step 1: Build a list with all the actions in the basic automata, $L$

    Step 2: For each action in the list $L$ get all the basic clauses associated
        with the action, building a vector $V$.

    Step 3: For each list of clauses in vector $V$, merge states: $merge(q_a, q_b)$

    Step 4: Create the product automaton.
```

Algorithm for merging states:

```
$merge(q_a, q_b)$:-
    Step 1: $q_a$ and $q_b$ are not from the same basic automaton

    Step 2: the actions of the states are the same, i.e., $\lambda(q_a) = \lambda(q_b)$.
    Step 3: Transitions are the same or not compatible
        i)  Same transitions, for any outcome $z$
            $q_{j+1} = \delta(q_a, z) = \delta(q_b, z)$

        ii) not(compatible($Z_a$ , $Z_b$)), in which $Z_a$ and $Z_b$ represent the
            transitions from states $q_a$ and $q_b$, respectively.
```

Proposition 5.5: *For the option product, as defined in Definition 5.2, to be a well defined finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *the following conditions are required*:

a) For all $q_{j1}^i \in Q^i$ *and* $\theta_{j1}^i \in \Sigma^i$*, such that* $q_{j1}^i \in Q_{j1}^i$ *, if* $\theta_{j1}^i = \lambda^i\left(q_{j1}^i\right)$ *then* $\theta_{j1}^i \in \Sigma_{j1}^i$.

b) Every transition from any state in a given option must be mutually excluding or lead to the same state. For every state q^i*, and outcomes* z_1^i, z_2^i*, if for at least one instance of* z_1^i, z_2^i *they are both true, i.e.,* $z_1^i \cap z_2^i$ *is true, then* $\delta^i\left(q^i, z_1^i\right) = \delta^i\left(q^i, z_2^i\right)$.

Proof: *a*) This proof is the same as in Proposition 5.1.*b*). *b*) Proof is the same as in Proposition 5.1.*c*). Q.E.D.

In Table 5.2 we present an algorithm to compute product options.

Table 5.2 The Product Option Algorithm

Parameters:

B: Number of basic automata

A^k: The basic automaton *k*, $A^k = \left(Q^k, q_0^k, \Sigma^k, Z^k, \delta^k, \lambda^k\right)$

Compatible outcomes: $Compatible\left(Z_w, Z_k\right)$: $Z_w \cap Z_k$ is *not* an impossible event

Compatible actions: $Compatible\left(a_w, a_k\right)$: $a_w \cap a_k$ is *not* an impossible event

$clause\left(k, q_j^k, a_j^k, z_j^k, q_{j+1}^k\right)$: stands for a clause in the basic option *k*, in which:

a) q_j^k : current state of the basic option

b) q_{j+1}^k: next state of the basic option if this clause is true

c) a_j^k represents the action at state q_j^k

d) z_j^k represents the outcome at state q_j^k

$clause\_product\left(q_j^p, a_j^p, z_j^p, q_{j+1}^p\right)$: stands for a clause in the product option, in which:

a) q_j^p, such that $q_j^p = \prod_{k=1}^{B} q_j^k$: current state of the product option

b) q_{j+1}^p, such that $q_{j+1}^p = \prod_{k=1}^{B} q_{j+1}^k$: next state of the product option if this clause is true

c) $a_j^p = a_j^1 a_j^2 ... a_j^B$, or equivalently $a_j^p = \prod_{k=1}^{B} a_j^k$, represents the product of the actions of the basic automata

d) $z_j^p = z_j^1 z_j^2 ... z_j^B$, or equivalently $z_j^p = \prod_{k=1}^{B} z_j^k$, represents the product outcome of the basic automata

Algorithm for the product automaton:

```
Let $P$ stand for the set of all game clauses of the product automaton
Initiate $P$ :={}
A state of the product automaton at time $j$ is $q_j^p = [q_j^1, q_j^2, ..., q_j^B]$
    or equivalently, $q_j^p = \prod_{k=1}^{B} q_j^k$ .
States_list :={}, represent the list of states to be analyzed

Step 1: Build the initial state of states
    a)    $q_0^p = \prod_{k=1}^{B} q_0^k$ or equivalently $q_0^p = [q_0^1, q_0^2, ..., q_0^B]$.
    b)    States_list := { $q_0^p$ }.

Step 2: While there are states in States_list, choose the first state in the list and
        compute the transition of the product automaton and add it to $P$

    a) States_list := {$S_1$| $Rest$}: choose the first state

    b) Expand($S_1$, New_clauses, New_states): expand the chosen state
    computing the transitions from that state and the New_states

        i) For each basic automaton $k$, at iteration $j$:
                Get $q_j^k$
                $a_j^k = \lambda(q_j^k)$

        ii) For each player $k$:
                $z_j^k = Z^k(a_j^1, a_j^2, ..., a_j^B)$
                $q_{j+1}^k = \delta(q_j^k, z_j^k)$

        iii) New state: $q_{j+1}^p = [q_{j+1}^1, q_{j+1}^2, ..., q_{j+1}^B]$

        iv) New real product option clause:
        $clause\_product(q_j^p, a_j^p, z_j^p, q_{j+1}^p)$
            $a_j^p = \prod_{k=1}^{B} a_j^k$
```

$$z_j^p = \prod_{k=1}^{B} z_j^k$$

c) Add the new *clause_product* to the product automaton

$$G := G \cup \{clause\_product : new\, clauses\, expanded\;\; from\; S_1\}$$

d) Add the new states to the list of states to be expanded and remove the first state:

i) $States\_list := \mathrm{Re}\,st \cup New\_states$
ii) Go to Step 2.a).

Next, in Figure 5.2 we present a simple example of an option product, in which a firm holds the options to invest in two different products *1* and *2*.

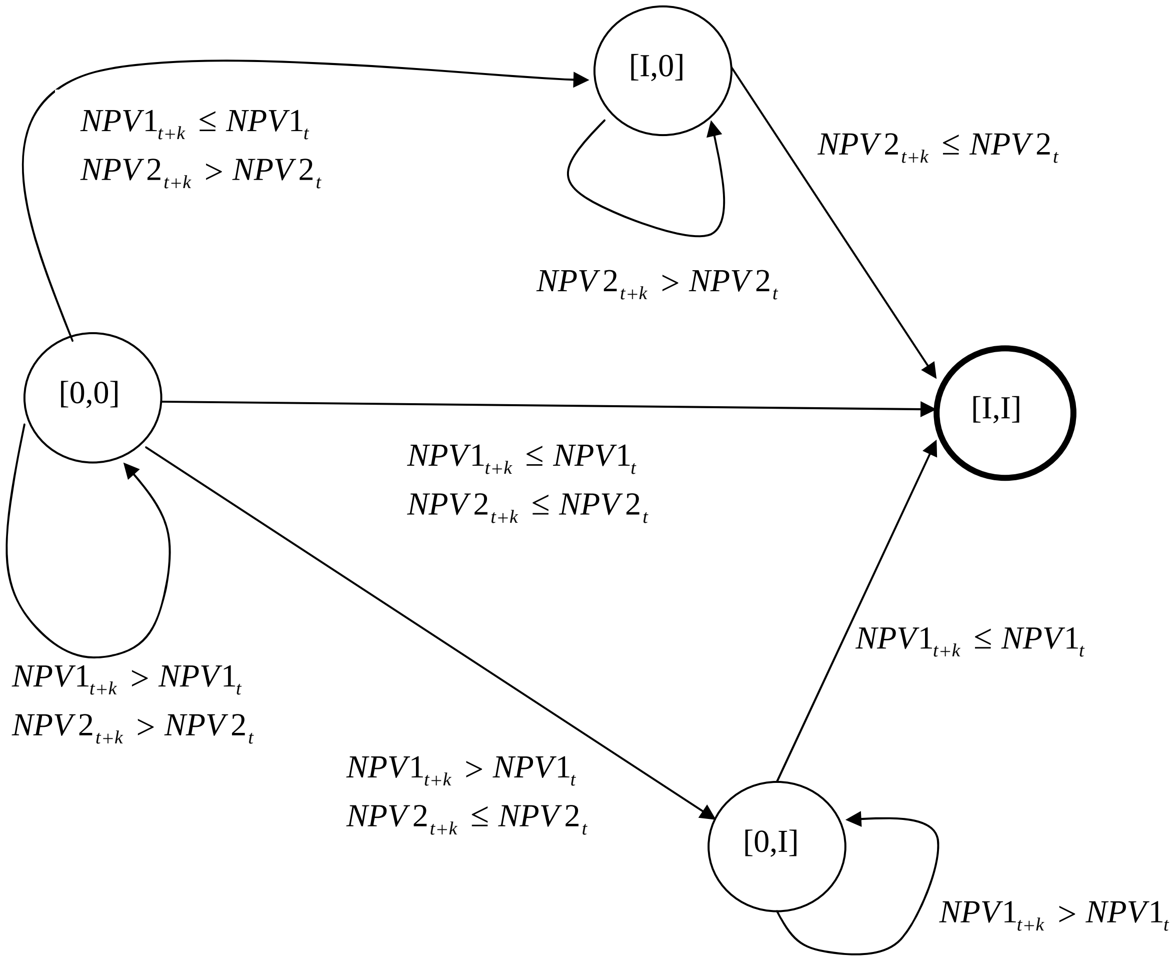

Figure 5.2: Product Automaton describing two different Options to Defer

In the product automaton in Figure 5.2 the number of states is the product of the number of states of the simple automata. The automaton starts in state [0,0] in which no option is exercised, and depending on the value of the net present values of each option the company may decide to go to state [I,0] or [0,I] in which one option is exercised, to continue in state [0,0] in which no option is exercised or move to state [I,I] in which both options are exercised. Only state [I,I] is absorbing.

6. Designing Real-Option Games

In this section, we look at the development of real-option games, in which the outcome of any real option depends on the real options of the other players in the game. Smit and Ankum (1993) presented one of the first models of real-options in multi-firm environments, namely modelling the entry game in a duopoly situation. In Table 6.1 we present the entry game in the normal form.

A firm *A* and a firm *B* need to decide when to undertake an investment. Let V_t represent the present value of the total cash flows the project generates when both firms invest simultaneously, and *I* the investment needed to undertake the given project. Then C_t represents the value of the option to invest, i.e., the value that the investment is expected to have if postponed. The model assumes that the company investing first earns a premium, the value of this investment will be V^{leader}_t, where $V^{leader}_t > V_t$. The value of the option to postpone an investment will also decrease if the other firm has invested first, $C^{follower}_t < C_t$.

Each company decides between investing and delaying the project. Under scenario *i* both firms decide to invest simultaneously and will get V_t-I. Under scenario *ii* (*iii*) the firm *A* (*B*) invests receiving the leader's payoff while the firm *B* (*A*) defers the investment and owns an option to invest that is less valuable. Scenario *iv* represents a situation in which both firms defer the investment owning, each one of them, an option with a value C_t.

Table 6.1: The entry game in the normal form

		Firm B	
		Invest	Defer
Firm A	Invest	i V_t-I, V_t-I	ii $V^{leader}{}_t$-I, $C^{follower}{}_t$
	Defer	iii $C^{follower}{}_t$, $V^{leader}{}_t$-I	iv C_t, C_t

A firm will invest at a period *t* if the NPV_t (V_t-*I* or $V^{leader}{}_t$-*I*) is higher than the value of postponing the investment, C_t or $C^{follower}{}_t$. In order to analyse the automaton for each player *A* and *B* let us simplify the representation of the several possible outcomes for the entry game. $D_F : V_t - I \le C_t^{follower}$; $I_F : V_t - I > C_t^{follower}$; $D_L : V_t^{leader} - I \le C_t$; $I_L : V_t^{leader} - I > C_t$.

The entry game is a simultaneous moves game in which the payoffs received by the players depend on the state of the industry and all the competitor's moves. The payoffs received by the players are the result of the Nash equilibrium: there is Nash equilibrium when no player can increase his payoffs by unilaterally changing his strategy. For this reason, there are two main issues with the entry game: a) as exposed in Smit and Ankum (1993) there are situations in which the payoffs received by the firms in the Nash equilibrium are lower than in the situation in which they coordinate their actions, this is known as the prisoners' dilemma; b) there may be a coordination problem if there are several Nash equilibria. This would be the case if $V_t - I \le C_t^{follower}$ as the value of the option to invest as a follower is not less than the value of investing simultaneously with the opponent firm. It is a natural assumption of the entry game for the value of the option to invest as a follower to be less than the value to invest as a Duopolist. For this reason this coordination problem is not an issue in the entry game.

The entry game in the automata format is represented in Figure 6.1 which represents the automata for firms *A* and *B*. An agent chooses to defer if the expected net present value of the investment is lower than the expected value of the call option, it will invest otherwise. The uncertainty regarding the decision to invest is on the other agent behaviour. The final result clearly depends on the expectations that each player holds about the others' behaviour.

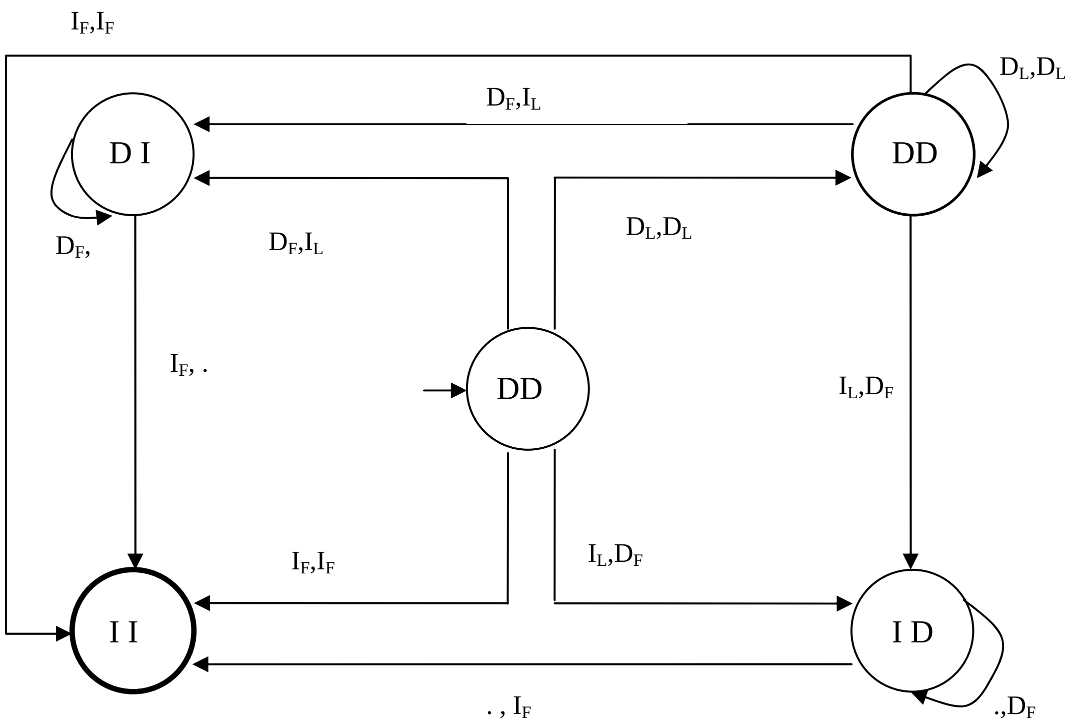

Figure 6.1: Automaton for the Entry Game

We are now in a condition to generalise the entry problem to *N* players using finite automata. In this setting we will assume that each agent knows the other players' automata (and their current internal state). Let us define the entry game in an industry with *N* potential firms. As in the two player entry game each firm decides *when* to enter the industry by making an irreversible investment. At each time each firm decides between investing or delaying the investment. The present value of the investment is a function of the number of companies in the market, i.e. $V_t = V_t(n)$, with $n=1,\ldots, N$, in which *n* represents the number of companies that have already invested (V_t is a negative function of *n*). The value of the call option associated with the opportunity to invest will also depend on the number of companies that have already invested: $C_t = C_t(n)$ with $n=0,\ldots, N-1$, (C_t is a negative function of n). Finally, we generalise the entry model assuming that the investment cost is also a function of the number of firms that have already invested: $I_t = I_t(n)$, (I_t is a negative function of n). For each player j, and at any time t, the payoff will be depend on the investment decision of the player j:

$$payoff_{jt} = \begin{cases} V_t(n) - I_t(n) \Leftarrow j \text{ invested} \\ C_t(n) \Leftarrow j \text{ not invested} \end{cases} \qquad \forall t, j.$$

Next, in Definition 6.1 we formalize the automaton used by a player i during the entry game.

Definition 6.1: *In the entry game each player uses an option to Defer which can be represented by a finite automaton* $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$ *such that: each element of the set of states* $Q^i = \{0,1,\ldots,N-1\}$ *equals the number of firms already in the market;* $q_0^i = 0$*;* $\Sigma^i = \{I, NI\}$ *is the set of all the possible actions, Invest (I) and delay (NI), and in which* $Z^i = \{W\text{:number of players entering the market at any time, } W \le N\text{-}1\}$ *represents the set of possible outcomes. Further, for any* $D_j \in Q^i$ *the transition function is equal to* $\delta^i\left(D_j, W\right) = D_j + W$*. Finally,* λ^i *is a behavioral function such that for any state* D_j*:* $\lambda^i\left(D_j\right) \in \{I, NI\}$.

As proved in Proposition 6.1, the Option Game is an instance of a Product Option, therefore, the Option Game Algorithm presented in Table 6.2 (which constructs an option game starting with the automaton of each player) is very similar to the Product Option algorithm presented in Table 5.2. The option game algorithm captures bounded rationality as it models games played by automata, i.e., by players that use rules of behaviour to manage the interactions with other players. In general, these rules of behaviour can have emerged through learning and evolution, or may be defined by the ethos of the firm. Nonetheless, these rules of behaviour may also be computed has the Nash equilibrium (and therefore as the perfect foresight solution to the game), as was the case of the entry game. However, in general, these Nash equilibria may not exist, there may be many of them, or the computational complexity of computing such equilibria (Gilboa and Zemel, 1989) may prevent the firms to use them in strategy problems.

Table 6.2 The Option Game Algorithm

Parameters:

N: Number of players in the game

A^i: Player *i's* automaton, $A^i = \left(Q^i, q_0^i, \Sigma^i, Z^i, \delta^i, \lambda^i\right)$

$clause\left(i, q_j^i, a_j^i, z_j^i, q_{j+1}^i\right)$: stands for a clause in the basic option of player *i*, defining the rules of transition from state to state, in which:

a) q_j^i: current state of the player *i's* automaton

b) q_{j+1}^i: next state of player *i's* automaton

c) a_j^i represents the action of player *i* at state q_j^i

d) z_j^i represents the outcome received by player *i* at state q_j^i

$clause\_game\left(q_j^g, a_j^g, z_j^g, q_{j+1}^g\right)$: stands for a clause in the option game, defining the rules of transition from state to state, in which:

a) q_j^g, such that $q_j^g = \prod_{i=1}^{N} q_j^i$: current state of the option game

b) q_{j+1}^g, such that $q_{j+1}^g = \prod_{i=1}^{N} q_{j+1}^i$: next state of the option game if this clause is true

c) $a_j^g = a_j^1 a_j^2 ... a_j^N$, or equivalently $a_j^g = \prod_{i=1}^{N} a_j^i$, represents the product of the actions of players 1 to *N*

d) $z_j^g = z_j^1 z_j^2 ... z_j^N$, or equivalently $z_j^g = \prod_{i=1}^{N} z_j^i$, represents the product outcome of the actions of players 1 to *N*, in state q_j^g.

Algorithm for the real options game:

Let *G* stand for the set of all game clauses of the real options game

Initiate *G* :={}

A state of the game at time *j* is $q_j^g = \left[q_j^1, q_j^2, ..., q_j^N\right]$.

States_list :={}, represent the list of states to be analyzed

Step 1: Build the initial state and the list of states:

a) $q_0^g = \prod_{i=1}^{N} q_0^i$ or equivalently $q_0^g = \left[q_0^1, q_0^2, ..., q_0^N\right]$.

```
        b)      States_list := { $q_0^g$ }.

    Step 2: While there are states in States_list, choose the first state in the list and
            compute the transition of the product automaton and add it to G

        a) States_list := {$S_1$| *Rest*}: choose the first state

        b) Expand($S_1$, New_clauses, New_states): expand the chosen state
        computing the transitions from that state and the New_states

            i) For each player $i$ at iteration $j$:
                    Get $q_j^i$
                    $a_j^i = \lambda\left(q_j^i\right)$

            ii) For each player $i$:
                    $z_j^i = Z^i\left(a_j^1, a_j^2, ..., a_j^N\right)$
                    $q_{j+1}^i = \delta\left(q_j^i, z_j^i\right)$

            iii) New state: $q_{j+1}^g = \left[q_{j+1}^1, q_{j+1}^2, ..., q_{j+1}^N\right]$

            iv) New real option game clause: $clause\_game\left(q_j^g, a_j^g, z_j^g, q_{j+1}^g\right)$
                    $a_j^g = \prod_{i=1}^{N} a_j^i$
                    $z_j^g = \prod_{i=1}^{N} z_j^i$

        c) Add the new *clause_product* to the product automaton

            $G := G \cup \{clause\_product : new\ clauses\ expanded\ \ from\ S_1\}$

        d) Add the new states to the list of states to be expanded and remove the
        first state:

            i) $States\_list := Rest \cup New\_states$
            ii) Go to Step 2.a).
```

Proposition 6.1: *The option game algorithm generates a product option* $A^g = \left(Q^g, q_0^g, \Sigma^g, Z^g, \delta^g, \lambda^g\right)$.

Proof: In order to prove this we need to show that any automaton produce by this algorithm obeys all the conditions in Proposition 5.5. *a*) There is an initial state $q_0^g = \prod_{i=1}^{N} q_{0,.}^{i}$ which is the product of the initial states of the *N* players. *b-d*) The set of internal states $Q^g = \prod_{i=1}^{N} Q_.^i$, actions, $\Sigma^g = \prod_{i=1}^{N} \Sigma_.^i$ and outcomes, $Z^g = \prod_{i=1}^{N} Z_.^i$ are the product of the different states of each one of the players in the game. *e-f*) The same transition function, $\delta^g : Q^g \times Z^g \to Q^g$, and behavioral function $\lambda^g : Q^g \to \Sigma^g$ defined in Proposition 5.5 also apply here as the sets Q^g, Z^g, Σ^g present the same properties as the sets Q^i, Z^i, Σ^i in Proposition 5.5. Q.E.D.

Proposition 6.2: *In the Nash Equilibrium of the Option game each player chooses the option that is a best response to the Product option of the Options used by his opponents*.

Proof: By definition, in a Nash equilibrium of the automata game, the automata choose by any player *i* is such that A_i is the best response (strategy) to the automata used by the other players, i.e., $A_i = BR(A_1, A_2, ..., A_{i-1}, A_{i+1}, ..., A_N)$. Since from Proposition 6.1 we know that the option game defined by the interaction of different options is a Product option, by defining $A^{g-i} = \left(Q^{g-i}, q_0^{g-i}, \Sigma^{g-i}, Z^{g-i}, \delta^{g-i}, \lambda^{g-i}\right)$ as the product automaton resulting from the interaction of *i*'s opponents, then it follows that $A_i = BR(A^{g-i})$. Q.E.D.

7. Conclusions

Strategy is a very interesting area of research which demands the development of new algorithms to handle the challenge of modelling complex and evolutionary games. In this paper we showed that automata theory is a powerful tool for the formal modelling of strategic decision making, enabling the analysis of bounded rationality and behavioural issues and enabling the graphical representation, and straightforward computational simulation and solution, of hard strategic decision-making problems.

Further, we showed how to formalize difficult strategy decision problems using finite automata, and provide two algorithms (and analyse their properties) for designing option combinations and product options. Finally, we provide an algorithm for a special case of a product option: the real option games, which enable the analysis of complex games in which different real options interact.

References

Abreu, D. and Rubinstein A. 1988. The Structure of Nash Equilibrium in Repeated Games with Finite Automata. Econometrica, 56 (6), Nov., 1259-1281.

Banks, J. S. and Sundaram, R. K. 1990. Repeated Games, Finite Automata, and Complexity. Games and Economic Behavior, 2: pp. 97-117.

Bollen N. P. B. 1999. Real Options and Product Life Cycles. Management Science, 45 (5): 670 – 684.

Cortazar, G., Schwartz E. S., Salinas M. 1998. Evaluating Environmental Investments: A Real Options Approach. Management Science, 44 (8): 1059 – 1070.

Dixit, A. 1992. Investment and Hysteresis. The Journal of Economic Perspectives, 6 (1): 107 – 132.

Dixit, A. and Pindyck, R. S., 1994. Investment under Uncertainty. Princeton University Press, New Jersey.

Eliaz K (2003) Nash equilibrium when players account for the complexity of their forecasts. Games and Economic Behavior: 44, 286-310.

Gilboa, I. 1988, The Complexity of Computing Best-Response Automata in Repeated Games. Journal of Economic Theory, 45, pp. 342-352.

Gilboa, I. and Zemel, E. 1989. Nash and Correlated Equilibria: Some Complexity Considerations. Games and Economic Behavior, 1: pp. 80 – 93.

Gossner, O., Hernandez, P. 2003. On the Complexity of Coordination. Mathematics of Operations Research :28(1), 127-140.

Grenadier, S. R. 1996. The Strategic Exercise of Options: Development Cascades and Overbuilding in Real Estate Markets. Journal of Finance, 51 (5): 1653 – 1679.

Huchzermeier, A. and Loch, C. H. 2001. Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D. Management Science, 47 (1), 85 – 101.

Hopcroft JE, Ullman JD (1979) Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Massachusetts.

Kardia, B., and Ernst, R., 2001. An Economic Model for Evaluating Mining and Manufacturing Ventures with Output Yield Uncertainty. Operations Research, 49(5): 690-699.

Kenyon, C., and Tompaidis, S., 2001. Real Options in Leasing: The Effect of Idle Time. Operations Research, 49(5): 675-689.

Kulatilaka, N., Perotti E. C. 1998. Strategic Growth Options. Management Science, 44 (8): 1021 – 1031.

Lander, M. L., and Pinches, G. E. 1998. Challenges to the Practical Implementation of Modeling and Valuing Real Options. The Quarterly Review of Economics and Finance, 38, Special Issue: 537 – 567.

Luehrman, T. A. 1998. Strategy as a Portfolio of Real Options. Harvard Business Review, September-October: 89 – 99.

McDonald R. and Siegel D. 1986. The Value of Waiting to Invest. Quarterly Journal of Economics, 101 (4): 707 – 728.

McGrath R. G. 1999. Falling Forward: Real Options Reasoning and Entrepreneurial Failure. Academy of Management Review, 24 (1): 13 – 30.

Neyman, A. 1998. Finitely Repeated Games with Finite Automata. Mathematics of Operations Research: 23(3), 513-551.

Papadimitriou, C. H. 1992. On Players with a Bounded Number of States. Games and Economic Behavior, 4, pp. 122 – 131.

Piccione M (1992) Finite automata equilibria with discounting. Journal of Economic Theory: 56, 180-193.

Pindyck, R. 1991. Irreversibility, Uncertainty, and Investment. Journal of Economic Literature, 29 (3): pp. 1110 – 1148.

Rubinstein, A. 1986. Finite Automata Play the Repeated Prisoner’s Dilemma. Journal of Economic Theory, 39, pp. 83-96.

Smit, H. T. J. and Ankun, L. A. 1993. A Real Options and Game-Theoretic Approach to Corporate Investment Strategy Under Competition. Financial Management, Autumn: pp. 241 – 250.

Smit, H.T. J. and Trigeorgis, L.(2004) *Strategic Investment: Real Options and Games*. Princeton University Press.

Thompson, M., Davison, M., and Rasmussen, H., 2004. Valuation and Optimal Operation of Electric Power Plants in Competitive Markets. Operations Research, 52(4): 546-562.

Trigeorgis, L. 1998. Real Options, Managerial Flexibility and Strategy in Resource Allocation. Third Edition, The MIT Press.