Chapter 2

Models of Asset Prices

Recall that a lattice has three components \((V, E, p)\) where \(V\) is a set of vertices, \(E \subseteq V \times V\) is the set of edges, and \(p(u, v)\) denotes the probability of transitioning from node \(u\) to \(v\). This chapter deals with the connection between lattices and models of asset prices. We also introduce the concept of state prices and risk-neutral probabilities in the context of lattices. Moreover, the concept of Markov processes generating a model of asset prices is also discussed.

2.1 Lattices as model of asset prices

We will have to add a few more components to lattices to model asset prices. Suppose we have \(n\) assets \(A_1, \ldots, A_n\) and we are interested in the price of these assets at times \(0, 1, 2, \ldots, T\). Time \(T\) is called the time horizon. Next we show how lattices can be used to model asset prices. A lattice model of asset prices has five components \(LM = (V, E, p, time, price)\), where \(V, E,\) and \(p\) have the same meaning as before, \(time\) is a function from \(V\) to \([T]\) (where \([T]\) = \(\{0, 1, 2, \ldots, T\}\)), and \(price\) is a function from \(V\) to \(\mathbb{R}^n\). For \(v \in V\), \(time(v)\) corresponds to time of the node and \(price(v)\) is the vector of prices for assets \(A_1, \ldots, A_n\) at node \(v\). We also assume that \((u, v) \in E\) implies that \(time(v) = time(u) + 1\), i.e., time increases by one time unit along the edges, and there is an unique node \(root \in V\) such that \(time(root) = 0\) (\(root\) corresponds to the initial node). A node \(v\) such that \(succ(v) = \emptyset\)
(v has no successors) will be called a leaf. Frequently, leaf nodes will correspond to the time horizon. The reader should intuitively think of a lattice as a model of the market (with respect to the n assets) and nodes as the states of the world. Probabilities $p(u, v)$ represent the likelihood of transition from state $u$ to $v$. We will use this intuitive notion of a lattice throughout the book. Hence, we will sometimes refer to a lattice model of asset prices as a market model.

Next we discuss a specific representation for lattices. Each node of a lattice has following components associated with it:

- **time** and **price**.
  Time and the vector of prices of $n$ assets corresponding to this node.

- **succ**
  Linked list of successors of this node.

- There might be other problem specific information associated with a node.

A lattice is represented as an array of linked lists $L[0], \ldots, L[T]$, where $L[t]$ is the linked list of nodes corresponding to time $t$.

### 2.2 Paths and Monte Carlo Simulation

Suppose we are given a lattice $LM = (V, E, W, time, price)$. A sequence of nodes $(v_0, v_1, \ldots, v_k)$ is called **path** in $LM$ iff $(v_i, v_{i+1})$ is an edge (or $(v_i, v_{i+1}) \in E$) for $0 \leq i \leq k - 1$. A path $\pi = (v_0, v_1, \ldots, v_k)$ is called **complete** iff $\text{pred}(v_0) = \emptyset$ ($v_0$ is the root) and $\text{succ}(v_k) = \emptyset$ ($v_k$ is a leaf node). Given a path $\pi = (v_0, v_1, \ldots, v_k)$, probability $P(\pi)$ associated with the path $\pi$ is obtained by multiplying the probabilities on the edges traversed by the path, or formally speaking

$$P(\pi) = \prod_{i=0}^{k-1} p((v_i, v_{i+1}))$$

Algorithm given in Figure 2.1 generates a random path through the **lattice** using a uniform number generator (available in most programming languages). Reasoning about the algorithm is left as an exercise for the reader.
2.2. PATHS AND MONTE CARLO SIMULATION

\textbf{proc} nextPath()
\begin{align*}
\text{begin} & \\
& ( \text{Put the initial node on the path}) \\
& path = v_0 \\
& ( \text{The last node of the path}) \\
& endNode = v_0 \\
& ( \text{keep traversing the path until it is complete}) \\
& \textbf{while} \ succ(endNode) \neq \emptyset \ \textbf{do} \\
& \quad ( \text{generate a random number with uniform distribution}) \\
& \quad x = uniform(); \\
& \quad ( \text{Find which successor of } endNode \ \text{to append}) \\
& \quad succ(endNode) = \{u[1], \ldots, u[m]\} \\
& \quad \textbf{for} \ i := 1 \ \textbf{to} \ m \ \textbf{step} \ 1 \ \textbf{do} \\
& \qquad weights[i] = p(endNode, u[i]) \\
& \quad \textbf{od} \\
& \quad \text{index} = 1 \\
& \quad w = 0.0 \\
& \quad \textbf{while} \ (w + weights[index] < x) \ \textbf{do} \\
& \qquad w = w + weights[index] \\
& \qquad \text{index} = \text{index} + 1 \\
& \quad \textbf{od} \\
& \quad path = \text{concat}(path, u[index]) \\
& \quad endNode = u[index] \\
& \textbf{od} \\
& \text{return}(path) \\
\textbf{end}
\end{align*}

\textbf{Figure 2.1:} Generate a random path through a lattice. Each call to \textit{uniform} generates a random number uniformly distributed between 0 and 1. Moreover, subsequent calls to \textit{uniform} generate independent random numbers.
Exercise 2.2.1 Prove that \textit{nextPath} generates a path \( \pi \) with probability \( P(\pi) \).

Exercise 2.2.2 Assume each node of the lattice can have at most \( M \) successors. Moreover, assume that calls to \textit{uniform} are primitive, i.e., takes \( O(1) \) time. What is the time complexity of \textit{nextPath}?

Notice that a series of calls to \textit{nextPath} generates a sequence of random paths through the lattice. We will use this for option pricing on the lattice using Monte Carlo simulation. Consider the lattice shown in Figure 2.2. Suppose two calls to routine \textit{uniform} produce random numbers 0.3 and 0.6. The path that is traced out in the lattice is shown by dotted lines.

2.3 Markov Processes

In our discussion random variables will take values over finite sets. Assume that we have a finite set \( D \). A function \( f : D \rightarrow \mathbb{R}_{\geq 0} \) is called a \textbf{probability distribution} over \( D \) iff it satisfies the following
2.3. MARKOV PROCESSES

condition:

$$\sum_{y \in D} f(y) = 1$$

The space of probability distributions over the set $D$ is denoted as $P(D)$. A random variable $X$ taking values over set $D$ has a distribution $f_X \in P(D)$ associated with it, where $f_X(d)$ is the probability that the random variable $X$ will realize the value $d \in D$.

Throughout this discussion we will implicitly assume that we have a time horizon of $T$ and are not interested in any information beyond time $T$. A process $\mathcal{P}$ over the set $D$ is a sequence of random variables $X_t$ and functions $F_t$ (the process is denoted as $\{X_t, F_t\}_{t=0}^T$) such that $X_0$ is an element of $D$ (a constant) and the distribution of $X_t$ is given by a function $F_t : \mathbb{R}^{t-1} \rightarrow P(D)$. $F_t(x_0, \cdots, x_{t-1})$ is the distribution of the random variable $X_t$ given that values of random variables $X_0, \cdots, X_{t-1}$ were $x_0, \cdots, x_{t-1}$. $X_t$ denotes the random value of the process at time $t$. In other words, the distribution of the random variable $X_t$ depends on the past history or the values of the random variables $X_0, \cdots, X_{t-1}$ according to the recipe $F_t$. Hence, a process is completely specified by an initial value $(X_0)$ and a recipe $(F_t)$ which provides a distribution of the values of the process at time $t$ based on the past history. Various classes of processes can be obtained by limiting the form of the function $F_t$.

Intuitively a process $\{X_t, F_t\}_{t=0}^T$ is called Markov iff distribution of $X_t$ only depends on the immediate history or the value of the random variable $X_{t-1}$. Formally, a process $\{X_t, F_t\}_{t=0}^T$ is called Markov iff the function $F_t(x_0, \cdots, x_{t-1})$ depends only on the variable $x_{t-1}$, or $F_t : \mathbb{R} \rightarrow P(D)$. A Markov process is called stationary or time homogeneous iff $F_t$ are all the same, i.e., there exists a function $F : \mathbb{R} \rightarrow P(D^n)$ such that $F_t = F$ for all $t \geq 1$. In other words for a stationary Markov process the recipe by which the distribution is specified does not depend on time. We will write a time homogeneous Markov process as $(\{X_t\}_{t=0}^T, F)$. Let us say we are given a stationary Markov process $(\{X_t\}_{t=0}^T, F)$ and suppose $F(x)$ is same for all $x$ or in other words $X_1, \cdots, X_T$ are independent random variables. Notice that in this case the distribution of $X_t$ does not depend on the history at all. In this situation we call the Markov process an iid process (iid
stands for independent identically distributed). An iid process
is written as \( \{X_i\}_{i=0}^{\infty}, P \) where \( P \in P(D) \). In case of an iid process
its value at time \( t \) is independent of its history and \( P \) specifies the
distribution of the process at any time between 1 and \( T \).

1-dimensional Random Walk

A 1-dimensional random walk with up-tick probability \( p \) (denoted as
randomWalk(1, \( p \)) is an iid process \( \{X_i\}_{i=0}^{T}, P \), where \( P \) is the fol-
lowing distribution:

\[
P(x) = \begin{cases} 
  p & \text{for } x = +1 \\
  1 - p & \text{for } x = -1 
\end{cases}
\]

Notice that in this case the range of random values or the set \( D \) is
\( \{+1, -1\} \).

\( d \)-dimensional Random Walk

Intuitively, a \( d \)-dimensional random walk with up-tick probability vector
\( (p_1, \cdots, p_d) \) is a vector of \( d \) independent 1-dimensional random walks

\[
\{X_{i,1}, \cdots, X_{i,d}\}_{i=0}^{\infty}, P
\]

where the probability of an up-tick of the \( j \)-th component of the random
walk \( X_{i,j} \) is \( p_j \). We denote this multi-dimensional random walk as
randomWalk(\( d, p_1, \cdots, p_d \)). In this case the random variables take value
in the set \( \{+1,-1\}^d \) or \( D = \{+1,-1\}^d \).

Exercise 2.3.1 Consider a 2-dimensional random walk \( \{X_{i,1}, X_{i,2}\}_{i=0}^{\infty}, P \)
with up-tick probability vector \( (p_1, p_2) \), or randomWalk(2, \( p_1, p_2 \)). De-
fine \( S_n = \sum_{i=1}^{\infty} (X_{i,1} + X_{i,2}) \). Derive a formula for \( P(S_n = k) \).

2.4 Markov Processes and Asset Prices

Suppose we have a Markov process \( \mathcal{P} = \{X_t, F_t\}_{t=0}^{\infty} \) and \( n \) assets
\( A_1, \cdots, A_n \). Markov process \( \mathcal{P} \) is said to drive the prices of the assets
\( A_1, \cdots, A_n \) if the price of the assets evolve in the following manner:
2.4. **MARKOV PROCESSES AND ASSET PRICES**

- Assume that the initial price of the assets is given by a vector \( \bar{x}_0 \).

- We assume that there is a function \( \text{price} : (D \times \mathbb{R}^n) \rightarrow \mathbb{R}^n \) where \( D \) is the domain associated with the process \( \mathcal{P} \). Interpretation of \( \text{price} \) is as follows: suppose we are at time \( t - 1 \) and the current price of assets is \( \bar{x} \) and the random variable \( X_t \) takes the value \( d_t \in D \), then the new price vector is given by \( \text{price}(d_t, \bar{x}) \). In general price of assets could also depend on time \( t \), and in this case we will write the price function as \( \text{price}_t \). In other words, the only randomness in the prices of the assets comes from process \( \mathcal{P} \).

Next, we show how to generate a lattice representation of prices of assets from the description of the Markov process \( \mathcal{P} \) that drives the asset prices. Assume that we are given a Markov process \( \{X_t, F_t\}_{t=0}^T \), \( n \) assets \( \{A_1, \cdots, A_n\} \), initial price vector \( \bar{x}_0 \), a price function \( \text{price} \) describing the evolution of asset prices, and a time horizon \( T \). We will give an algorithm to generate lattice which models the prices of the assets \( \{A_1, \cdots, A_n\} \).

A detailed explanation of the algorithm given in Figure 2.3.

- Each node has three components \([t, y, \bar{x}]\), where \( t \) represents the time, \( y \) represents the value of the random variable \( X_t \) at time \( t \) and \( \bar{x} \) is the price of the asset at time \( t \). Basically, we keep track of time, value of the driving stochastic process, and asset prices at each node. Notice that if the driving Markov process \( \{X_t, F_t\}_{t=1}^T \) is iid (like in the binomial model) we do not need the second component because the distribution of \( X_{t+1} \) does not depend on the value of \( X_t \).

- Queue \textit{unexplored} contains the set of nodes that have been generated but that have not been explored.

- At each step in the algorithm we take an unexplored node \( n \) off the queue \textit{unexplored}. If node \( n \) corresponds to time horizon \( T \), we do nothing. Otherwise we generate all successors of \( n \), and fill in the required edges. The reader is encouraged to identify the lines in the pseudo-code that correspond to these steps.
**proc** GenLattice($\{X_i, F_i\}_{i=0}^T, \tilde{x}_0, \text{price}, T$)

begin
( Queue of unexplored nodes)
unexplored := \emptyset;
( Generate the initial node and put it in the unexplored queue)
EnQueue(unexplored, [0, X_0, \tilde{x}_0]);
while not Empty(unexplored) do
( Explore node)
n := [t, y, p] := DeQueue(unexplored);
if \(t < T\) Have we reached the time horizon?
then distribution = \(F_{t+1}(y)\)
for \(z \in D\) do
\(x_{\text{new}} = \text{price}(z, \tilde{x})\)
\(n_{\text{new}} := [t + 1, z, x_{\text{new}}]\)
addEdge$(n, n_{\text{new}}, \text{distribution}(z))$
( Check if new node already there)
if \(n_{\text{new}} \notin \text{unexplored}\) then put(unexplored, \(n_{\text{new}}\)) fi
od
fi
od
end

Figure 2.3: Generate the lattice that models the asset price values
The Binomial model

Here we describe the algorithm $GenLattice$ as it applies to the binomial model. In this case the driving process is the 1-dimensional random walk $randomWalk(1, p)$, which was described earlier. Recall that $randomWalk(1, p)$ is an iid process $\{X_t\}_{t=0}^\infty$ such that $X_0 = 0$, all $X_t$ are independent, and $X_t$ is a random walk with up-tick probability $p$. Distribution of $X_t$ is as follows:

$$X_t = \begin{cases} +1 & \text{with probability } p \\ -1 & \text{with probability } 1 - p \end{cases}$$

Assume that we have one asset $S$ called the stock. The price function price is

$$price(x, s) = \begin{cases} su & \text{if } x = +1 \\ sd & \text{if } x = -1 \end{cases}$$

In other words, the stock price goes up by a factor $u > 1$ if the random variable $X_t$ realizes the value of $+1$ and goes down by a factor $-1$ otherwise. We will use the algorithm $GenLattice$ to generate the binomial model for two time periods or $T = 2$. Initial stock price is $s_0 = 1$.

- Initially the node $[0, 1]$ is put on the queue unexplored.
- Node $[0, 1]$ is taken off the queue unexplored and its successors $[1, u]$ and $[1, d]$ are generated and added to the queue unexplored. Edges $([0, 1], [1, u])$ and $([0, 1], [1, d])$ are added to the lattice with probabilities $p$ and $1 - p$.
- Next a node $[1, u]$ is taken off the queue unexplored. Its successors are $[2, u^2]$ and $[2, ud]$ and added to the queue unexplored. Next, node $[1, d]$ is taken off the queue unexplored and its successors $[2, ud]$ and $[2, d^2]$ are generated. Notice that $[2, ud]$ is already in the queue unexplored, so it is not put in the queue. Node $[2, d^2]$ is put in the queue. Nodes $[2, u^2]$, $[2, ud]$ and $[2, d^2]$ have no successors (because the time horizon is $T = 2$) so they are taken off the queue and the algorithm ends.

The algorithm $GenLattice$ works forward in time. This is known as forward induction and later we will discuss this concept in great detail.
Next we analyze the space complexity of the binomial model with time horizon $T$. Let $D_t$ be the number of down-ticks that occur up-to time $t$, i.e., size of the following set
\[ \{i | 1 \leq i \leq t, \ X_i = -1\} \]
The number of up-ticks that occur up-to time $t$ (denoted by $U_t$) satisfies the following equation:
\[ U_t = t - D_t \]
Let $S_t$ be the stock price at time $t$. From the equation that describes the evolution of the stock price, one can easily deduce that $S_t$ has the following form:
\[ s_0 u^{U_t} d^{D_t} \]
In the equation given above $s_0$ is the initial stock price. Since $0 \leq D_t \leq t$, there are $t + 1$ possible stock prices at time $t$ (notice that fixing the number of down-ticks fixes the number of up-ticks, so there is only one free variable). The possible number of stock prices up-to time $T$ is given by the following equation:
\[ 1 + 2 + \cdots + T + (T + 1) = \frac{(T + 1)(T + 2)}{2} \]
Hence, the space complexity of the binomial model is $\Theta(T^2)$ or quadratic in the time horizon.

**Maximum of the Stock Price**

There are several options (like lookback) whose payoff at a certain time depends on the maximum of the stock prices seen up-to that time. We will show how to generate a lattice that allows one to price these kind of options. Intuitively, we have to keep track of the stock price as well as the maximum of the stock prices seen so far. We use the binomial model for the stock prices. Now there are two assets: stock price $S$ and the maximum $M$ of the stock prices seen so far. The asset prices are given by a vector of size 2. The price function $\text{price}$ is given as follows:
\[
\text{price}(x, (s, m)) = \begin{cases} 
(su, \max\{m, su\}) & \text{if } x = +1 \\
(sd, \max\{m, sd\}) & \text{if } x = -1
\end{cases}
\]
2.4. MARKOV PROCESSES AND ASSET PRICES

In the equations given above the first component refers to the stock price and the second to the maximum of the stock prices seen so far.

**Pricing a general option on a stock**

Assume that we define a process $K$ derived from the stock process $S$ as follows:

$$\text{new}(K) = F(\text{old}(K), \text{new}(S))$$

In other words we are given a function $F$ of two variables and the new value of the process $K$ is calculated by using the old value of the process $K$ the new stock price as the first and the second parameters respectively. Recall that in the case of when $K$ was the maximum of the stock prices seen so far (used to price the lookback option) the function $F$ had the following form: $F(x, y) = \max(x, y)$. Consider the following function:

$$F(x, y) = x \cdot y$$

Let us see what is the process that $F$ corresponds to. $S_t$ is the stock price at time $t$. Suppose at the initial node the value of this mystery process is $S_0$. Next period the value of the process is $S_0 S_1$. In the second period it is $S_0 S_1 S_2$. The value of the mystery process at time $t$ is $S_0 S_1 S_2 \cdots S_t$. If the value of this mystery process is $c$ at time period $t$, then $c^{\frac{1}{t+1}}$ is the geometric average of the stock price process. In order to price an option which is based on the geometric average we have two assets $S$ (the stock price) and the asset $GA$ such that the price function $price$ is given as follows:

$$price(x, (s, g)) = \begin{cases} 
(su, gsu) & \text{if } x = +1 \\
(sd, gsd) & \text{if } x = -1
\end{cases}$$

Given an option, one should try to choose an auxiliary process so that given the value of the stock and the auxiliary process the payoff of the option is completely determined. In other words, we expand the state space of the binomial model such that the payoff function of the option becomes Markovian, i.e., is completely determined by the state or
node of the constructed model. Notice that the function \( F \) completely
determines the evolution of the auxiliary process, or in other words the
auxiliary process is also Markovian, i.e., is completely determined by
its old value and the new value of the stock price. All options cannot
be priced using this general methodology. We will see some examples
later.

**Exercise 2.4.1** Consider the binomial model of stock price with \( u = \frac{1}{2} \).
At time \( t \) how many possible geometric averages of the stock prices
are there. **Hint:** Stock price \( S_t \) at time \( t \) is given by the following
expression

\[
S_0 u \sum_{i=1}^{t} x_i
\]

Let \( G_t \) be the geometric average at time \( t \). \( G_t \) is described by following
series of equations:

\[
G_t = S_0 S_1 S_2 \cdots S_t
\]

\[
= S_0^{t+1} u \sum_{i=1}^{t} \sum_{i}^{t} x_i
\]

\[
= S_0^{t+1} u \sum_{i=1}^{t} (t-i+1) x_i
\]

Now find the maximum and the minimum value that \( G_t \) can take. The
difference between the maximum and the minimum value is an upper
bound on the possible number of values for \( G_t \).

**Exercise 2.4.2** There are some options (for example *asian*) where the
payoff depends on the arithmetic average of the stock prices. Pick an
appropriate function \( F \) and show the generated lattice for two periods.

### 2.5 Arbitrage, complete markets, and state prices

This section discusses the important concepts of arbitrage, complete
markets and state prices. Assume that we have \( n \) assets \( \{A_1, \cdots, A_n\} \)
and a lattice

\[
LM = (V, E, W, time, price)
\]

model of asset prices at times \( \{0, 1, 2, \cdots, T\} \). Recall that \( time \) is a function
from \( V \) to \([T]\) (where \([T]\) = \( \{0, 1, 2, \cdots, T\} \)) and \( price \) is function
2.5. ARBITRAGE, COMPLETE MARKETS, AND STATE PRICES

from $V$ to $\mathbb{R}^n$. If for $v \in V$, $(time(v), price(v)) = (t, \bar{x})$, node $v$ corresponds to time $t$ and $\bar{x} \in \mathbb{R}^n$ is the vector of prices of assets $A_1, \cdots, A_n$ respectively. Also, recall that there is an unique node (called root) such that $time(root) = 0$ (the node root corresponds to the initial time). A portfolio process $\Delta$ is a function from $V$ to $\mathbb{R}^n$. $\Delta(v)[i]$ is the number of units of asset $A_i$ held at node $v$. Let us assume that we start with initial wealth $X_0$ and $X_0 = \Delta_0 \cdot price(root)$ (value of the portfolio at the root is equal to the initial wealth). Since $\Delta_0$ and $price(root)$ are vectors of size $n$, $\cdot$ denotes the dot product. Wealth at other nodes is simply the value of the portfolio at that node, or is given by the following equation:

$$X(v) = \Delta(v) \cdot price(v)$$

Consider a node $v$ and one of its successor $v'$. The value of the portfolio $\Delta(v)$ at node $v'$ is

$$\Delta(v) \cdot price(v')$$

Portfolio $\Delta(v')$ at node $v'$ has the value

$$\Delta(v') \cdot price(v')$$

The difference between the values of the portfolios $\Delta(v)$ and $\Delta(v')$ is consumed in going from node $v$ to $v'$. Let $consume(v, v')$ denote the consumption from node $v$ to $v'$. $consume(v, v')$ is given by the following equation:

$$consume(v, v') = price(v')(\Delta(v) - \Delta(v'))$$

Consumption $consume(v, v')$ in going from node $v$ to $v'$ can be negative (infusion of cash), zero, or positive (consumption of cash).

**Definition 2.5.1** A portfolio process $\Delta$ is called an arbitrage if and only if it satisfies one of the following conditions:

- For the root node $v$, $\Delta(v) \cdot price(v) < 0$, $consume(v, v') \geq 0$ for all edges $(v, v')$, and for each leaf node $v$, $X(v) \geq 0$. The investor taking advantage of this condition can receive some money at the beginning ($\Delta(v) \cdot price(v) < 0$), needs no infusion of cash ($consume(v, v') \geq 0$), and at the end is left with non-negative wealth ($X(v) \geq 0$).
• For the root node $v$, $\Delta(v) \cdot \text{price}(v) = 0$, $\text{consume}(v, v') \geq 0$ for all edges $(v, v')$, and for each leaf node $v$, $X(v) \geq 0$. Moreover, there exists at least one edge $(v, v')$ such that $\text{consume}(v, v') > 0$ or a leaf node $v$ such that $\Delta(v) \cdot \text{price}(v) > 0$. If the conditions stated above are true, an investor can start with zero initial capital and invest so that he/she needs no infusion of capital, and can either consume a positive amount somewhere, or is left at the end with a positive wealth.

Intuitively, an arbitrage is an investment strategy which does not need any money up-front but has a non-zero probability of making money, or \textit{chance of making something for nothing}. A lattice is called \textbf{arbitrage free} iff it does not admit arbitrage.

Consider a lattice $LM = (V, E, W, \text{time}, \text{price})$. Given a node $v \in V$ such that $\text{succ}(v) \neq \emptyset$, let $LM_1(v)$ consist of $v$ and its successors. A node $v$ is called \textbf{arbitrage free} iff the lattice $LM_1(v)$ is arbitrage free. $LM_1(v)$ can be viewed as a lattice model of prices with time horizon 1. In general, a sub-lattice of $LM$ that is rooted at node $v$ and contains all descendants $v'$ of $v$ such that $\text{time}(v') - \text{time}(v) \leq k$ is denoted by $LM_k(v)$. $LM_k(v)$ is a lattice model of asset prices with time horizon $k$. $LM_k(v)$ is said to be a sub-lattice of $LM$ of depth $k$. The theorem given below states that in order to test that a lattice is arbitrage free it is enough to check that every sub-lattice of depth 1 is arbitrage free. This result greatly simplifies the computational complexity of testing that a lattice is arbitrage free.

\textbf{Theorem 2.5.1} Consider a lattice model $LM = (V, E, W, \text{time}, \text{price})$ of asset prices. Lattice $LM$ is arbitrage free iff every node of $LM$ is arbitrage free.

\textbf{Proof:} We will prove that every sub-lattice of depth $k + 1$ is arbitrage free iff every sub-lattice of depth $k$ is arbitrage free. Notice that this proves by transitivity that lattice $LM$ (of depth $T$) is arbitrage free if every sub-lattice of depth 1 is arbitrage free (or equivalently every node is arbitrage free).

\textbf{Exercise 2.5.1} Assume that you are given a lattice $LM = (V, E, W, \text{time}, \text{price})$. Give an algorithm to prove that every node of the lattice is arbitrage
free. Assume that you have a routine which tells you whether a system of inequalities is feasible. Analyze the complexity of your algorithm.

**State prices**

Suppose we are given a lattice $LM = (V, E, W, time, price)$ and a non-root node $v' \notin V$. $\delta(v')$ is called a **pure contingent claim** at node $v'$ iff it pays 1 at node $v$ and pays 0 everywhere else. Consider a node $v \in pred(v')$ ($v$ is a predecessor of $v'$). A **state price** along the edge $v \rightarrow v'$ is the unique value of the contingent claim $\delta(v')$ at node $v$, or the present value of the contingent claim.

**Exercise 2.5.2** Prove that if state prices along all the edges of a lattice $LM = (V, E, W, time, price)$ are positive, then $LM$ is arbitrage free.

The following theorem also proves the converse of exercise 2.5.2.

**Theorem 2.5.2** Given a lattice $LM = (V, E, W, time, price)$ that is arbitrage free, the state prices along all edges are positive.

Notice that the theorem given above says nothing about the uniqueness of the state prices. We proceed to address that question. Suppose we are given a lattice $LM = (V, E, W, time, price)$. We will show how to compute state prices. Consider a node $v$ such that $\text{succ}(v) \neq \emptyset$ (or $v$ has successors). Let $\Delta(v, v')$ be the set of portfolios $\Delta$ such that $\Delta \cdot \text{price}(v') = 1$ and $\Delta \cdot \text{price}(v'') = 0$ for all $v'' \in \text{succ}(v)$ not equal to $v$. In other words, $\Delta(v, v')$ is the set of portfolios that **replicates** the contingent claim $\delta(v')$.

**Exercise 2.5.3** Consider two portfolios $\Delta_1$ and $\Delta_2$ in the set $\Delta(v, v')$. Prove that in the absence of arbitrage $\Delta_1 \cdot \text{price}(v) = \Delta_2 \cdot \text{price}(v)$.

Define the state price on the edge $(v, v')$ (denoted by $\lambda(v, v')$) as $\Delta \cdot \text{price}(v)$ for an arbitrary $\Delta \in \Delta(v, v')$. Consider a node $v$ whose set of successors is $\{v_1, \ldots, v_k\}$. Consider an financial asset which pays $\text{payoff}(v_i)$ at node $v_i$. The unique price of this financial asset is:

$$
\sum_{i=1}^{k} \lambda(v, v_i) \text{payoff}(v_i)
$$
In other words, state prices allow us to determine payoffs of arbitrary financial assets or instruments.

Consider a node $v$ with $k$ successors $\{v_1, \ldots, v_k\}$, and a $k \times n$ matrix $nextPrice(v)$ such that $nextPrice(v)[i,j] = price(v_i)[j]$, i.e., price of $j$-th asset at node $v_i$. The equation given below has an unique solution iff $rank(nextPrice(v)) \geq k$.

$$M \cdot \Delta^T = e(1)$$

where $e(1)$ is the vector $(1, 0, \ldots, 0)$, i.e., the payoff of the contingent claim $\delta(v_1)$. Notice that that once we have solved the equation given above, the state price $\lambda(v, v_1)$ is given by $\Delta \cdot price(v)$. If $rank(nextPrice(v)) \geq k$, we can also solve for the following equation (for $1 \leq r \leq k$)

$$M \cdot \Delta^T = e(r)$$

where $e(r)$ is the vector of length $k$ whose $r$-th entry is 1 and all other entries are 0. Notice that the system of equations lets us calculate states prices $\lambda(v, v_r)$ for all $r$, i.e., all state prices along the edges emanating from $v$. In order for state prices to exist among all edges of a lattice $LM = (V, E, W, time, price)$, we should have the following inequality for each node $v$ with successors

$$rank(nextPrice(v)) \geq |succ(v)|$$

If the equation given above is true for each node $v$ with successors in a lattice $LM$, we will say that $LM$ represents a complete market.

**The risk-neutral measure**

Again assume that we are given a lattice $LM = (V, E, W, time, price)$ model of prices of assets $\{A_1, \ldots, A_n\}$. Assume that $A_1$ is risk free, or satisfies the following conditions:

- price of asset $A_1$ is positive at every node

- given a node $v$ with successors, price of the asset $A_1$ is the same at every successor of $v$, i.e., given two arbitrary successors of $v_1$ and $v_2$ of $v$ we have the following equality:

$$price(v_1)[1] = price(v_2)[1]$$
2.5. ARBITRAGE, COMPLETE MARKETS, AND STATE PRICES

Consider a node $v$ with $\{v_1, \cdots, v_k\}$ as its successors. Let $\text{price}(v_1)[1] = \cdots = \text{price}(v_k)[1] = w$. We have the following equation:

$$\text{price}(v)[1] = w \sum_{i=1}^{k} \lambda(v, v_i)$$

$$\sum_{i=1}^{k} \lambda(v, v_i) = \frac{\text{price}(v)[1]}{w}$$

Define $p(v, v_i)$ as

$$\lambda(v, v_i) = \frac{w}{\text{price}(v)[1]}$$

Notice that $\frac{w}{\text{price}(v)[1]}$ is the payoff at the next time on 1 dollar invested at node $v$ in the risk-less asset $A_1$. We will call this the discount rate $R$. The short rate $r$ is defined as $R - 1$. Hence the equation for $p(v, v_i)$ can also be written as

$$\lambda(v, v_i)(1 + r)$$

It can be easily seen that $\sum_{i=1}^{k} p(v, v_i) = 1$. If $A_1$ is a risk-less asset, then we can convert state prices into probabilities. These probabilities are called risk-neutral probabilities.

The binomial model

We consider the special case of the binomial model. Assume that we have two assets: a stock and bond. One dollar invested in a bond pays $1 + r$ in each of the next two states (where $r$ is the short rate). Stock prices are shown in Figure 2.4.

Let $\Delta_u$ be the portfolio which replicates the contingent claim $\delta(u)$, i.e., $\Delta_u$ satisfies the following equations:

$$\Delta_u[1](1 + r) + \Delta_u[2]s_0 u = 1$$

$$\Delta_u[1](1 + r) + \Delta_u[2]s_0 d = 0$$

In the equations given above, $\Delta_u[1]$ and $\Delta_u[2]$ represent the amount invested in bonds and the number of units of stock respectively. Solving the equations given above we obtain:

$$\Delta_u[1] = \frac{-s_0 d}{s_0(u - d)(1 + r)}$$
\[ \Delta_u[2] = \frac{1}{s_0(u - d)} \]

State price \( \lambda_u \) (along the edge \( (v, v_u) \)) is given by

\[ \lambda_u = \frac{(1 + r) - d}{(1 + r)(u - d)} \]

Similarly, state price \( \lambda_d \) (along the edge \( v, v_d \)) is given by:

\[ \lambda_d = \frac{u - (1 + r)}{(1 + r)(u - d)} \]

The risk-neutral probability \( p \) and \( q \) are given by the following equations:

\[ p = \frac{(1 + r) - d}{u - d} \]
\[ q = \frac{u - (1 + r)}{u - d} \]

**Change of numeraire**

Earlier we described how to derive risk-neutral probabilities from state prices if there exists a risk-less asset among the set of assets we are
2.5. ARBITRAGE, COMPLETE MARKETS, AND STATE PRICES

considering. What if there isn’t a risk-less asset in the set of assets being considered? We will show that is enough to have an asset whose price is positive at every node to derive risk-neutral probabilities. As usual assume that we have \( n \) assets \( \{A_1, \ldots, A_n\} \) and a lattice \( LM = (V, E, W, \text{time, price}) \) modeling the asset prices. Moreover, assume that price of asset \( A_1 \) is positive at every node. The basic idea is to change the numeraire to the asset \( A_1 \). This means all prices are expressed as number of units of asset \( A_1 \). So for example, \( \text{price}_1(v)[j] \) is the price of asset \( A_j \) in terms of how many units of asset \( A_1 \) can we buy. Following relationship between prices in dollars and the numeraire \( A_1 \) is easy to see:

\[
\text{price}_1(v)[j] = \frac{\text{price}(v)[j]}{\text{price}(v)[1]}
\]

Price of asset \( A_1 \) under this new numeraire is 1 everywhere.

Let \( \lambda_1(v, v') \) be the state price using the numeraire \( A_1 \). We have the following equation:

\[
\lambda_1(v, v') = \lambda(v, v') \frac{\text{price}(v')[1]}{\text{price}(v)[1]}
\]

In other words, the state price using the numeraire \( A_1 \) is the state price using the original numeraire multiplied by the ratio of the price of the asset \( A_1 \) in the next state and price of asset \( A_1 \) in the current state.

**Exercise 2.5.4** Prove equation 2.1. **Hint:** Go back to the definition of state prices as prices of pure contingent claims.

Suppose a node \( v \) has \( k \) successors \( \{v_1, \ldots, v_k\} \). We have the following equation:

\[
\sum_{i=1}^{k} \lambda_1(v, v_i) = \sum_{i=1}^{k} \lambda(v, v_i) \frac{\text{price}(v_i)[1]}{\text{price}(v)[1]}
\]

\[
= \frac{\sum_{i=1}^{k} \lambda(v, v_i) \text{price}(v_i)[1]}{\text{price}(v)[1]}
\]

\[
= 1
\]

Therefore, under the new numeraire state prices represent risk-neutral probabilities.
Exercise 2.5.5 Consider the example of a stock and a bond. This time use stock as the numeraire and derive state prices. Check equation 2.1. Prices of stock and bond under this new numeraire are shown in Figure 2.5.

Games with redundant securities

Suppose we are given a lattice model of asset prices and state prices along the edges of the lattice. Consider a node \( v \) with its sets of successors as \( \{v_1, \ldots, v_k\} \). Suppose we are given an asset and its price at nodes \( v_1, \ldots, v_k \). The price of this asset at node \( v \) is given by the following expression:

\[
\sum_{i=1}^{k} \lambda(v, v_i) z_i
\]

Price of the asset at node \( v_i \) is \( z_i \). Therefore, once we determine the state prices, the movement of prices along on the lattice model is restricted. We will provide an interesting consequence of this phenomenon.
2.5. Arbitrage, Complete Markets, and State Prices

![Figure 2.6: Evolution of exchange rate](image)

Figure 2.6: Evolution of exchange rate

Assume that we have a domestic economy (say the US) and a foreign economy. Consider a node $v$ with two successors $v_u$ and $v_d$. Let the exchange rate be $E$, $E_u$, and $E_d$ at the nodes $v$, $v_u$, and $v_d$ respectively (see Figure 2.6). The exchange rate determines the value of an unit of currency in the domestic economy in terms of the foreign currency. Let $r$ be the short rate in the domestic economy. The two independent assets are:

- Bond in the domestic economy.
- Domestic currency changed to foreign currency and then converted back to the domestic currency.

**Exercise 2.5.6** Derive the risk-neutral probabilities using the two assets just described.

**Exercise 2.5.7** Let $r_f$ be the short rate in the foreign economy. Suppose we have 1 unit of the domestic currency. Changing this to foreign currency we have $E$ units in the foreign currency. Investing this in the foreign bond market provides us with the payoff of $\frac{E(1+r_f)}{E_u}$ and $\frac{E(1+r_f)}{E_u}$.
at the nodes $v_u$ and $v_d$. Therefore, the price of this asset can be determined using the risk-neutral probabilities computed in exercise 2.5.6. This price must be equal to 1 (why?). This gives us an equation between the domestic short rate, the foreign short rate, and the exchange rate.