Chapter 1
Games, logic and automata

Game Theory is the study of situations where people with conflicting or collective interests make decisions to achieve certain outcomes. A game is traditionally a collection of strategies (how to play) for the players of the game with each player having preferences over various outcomes that are generated by tuples of strategies. One of the main aims of game theoretic analysis is to predict what outcomes would result if people play according to certain ‘rationality’ assumptions. In other words, given a game situation, we would like to predict ‘equilibrium’ or ‘stable’ play.

One of the many definitions of equilibrium play requires that players should not have an incentive to deviate from such a play. Towards this end, John Nash [Nas50] in his path breaking work showed that every finite game has an equilibrium, which is now known as the Nash equilibrium, from which no player has an incentive to unilaterally deviate. An equilibrium may be viewed as a ‘solution concept’ of a game.

Many attempts have been made at providing logical foundations to the various equilibria and other solution concepts. The fact that every finite game has a Nash equilibrium is quite encouraging. But why should players play for such an equilibrium? What is the reasoning required on their part to play a Nash equilibrium? Aumann and Dreze [AD05] point out that game theory started out by trying to develop a prescriptive theory for rational agents right from the seminal work of von Neumann and Morgenstern [vNM44] who envisaged game theory as constituting advice for players in game situations, so that strategies may be synthesised accordingly. Such a prescriptive theory must account for the beliefs and expectations each player has about the strategies of the other players.

John Harsanyi [Har77] in his book *Rational Behaviour and Bargaining*
Equilibrium in Games and Social Situations, developed a beautiful theory of ‘rationality’ and how ‘rational’ players actually behave. He argued that rational players play rationally and also believe that other players are rational. This means that they know that others know that they are rational, they know that others know that they know that the others are rational and so on. Such a hierarchy of knowledge in its infinite level is called the common knowledge of rationality. The existence of Nash equilibrium in a game in general requires the assumption that the players are rational and it is common knowledge that they are rational. Thus playing an equilibrium strategy requires players to reason based on how other players reason.

Such mutually recursive reasoning by the players can be justified when the games are small, in that, there are a small number of players or the structure of the games is small. However such justification rapidly breaks down when the games become larger and larger. We call a game large if it has one or both of the following features:

- A large number of players.
- A large temporal or spatial structure. That is, the game continues for a long duration, possibly unbounded, or the players are scattered across a large geographical or logical expanse. An example of a game satisfying the latter criterion is when A learns about X from her neighbour B who in turn learns about her from her own neighbour C and so on. Thus although A may not be geographically distant from X, she is logically so.

In such large games, various forms of uncertainties arise: players in one part of the game may be uncertain about the outcomes in some other part of the game, they may be uncertain about the strategies of the other players, even the number of players playing the game. We claim that such uncertainty can arise even in perfect information games of the classical sense. In such a situation, the traditional style of equilibrium analysis of games lack proper motivation. If the players do not even know how many others or who they are playing against, how are they to reason mutually recursively to attain equilibrium play?

Uncertainty in games has been handled traditionally with the concept of information sets. An information set of a player is a collection of game histories between which a player cannot distinguish with her observations so far in the game. Information sets model very well situations where players do not have complete information about the other players’ moves or strategies. However they, in general, do not deal with uncertainty of the kind that arise
in large games as described above. The definition of information sets can no
doubt be tweaked to take into account such uncertainties. For example, if a
player A is uncertain about the number of other players playing the game, at
every position, her information set would contain elements that correspond
to there being 2, 3, . . . players in the game. But then, such a model, among
others, has the immediate drawback of being infinite. Hence, we feel that
modeling such games using the notion information sets is unwieldy and
problematic.

Another important aspect to consider in real-life game-playing situations
is that the players have computational limitations (resources, memory etc.).
They cannot in general carry out the complicated reasoning to play an equi-
librium strategy. Moreover, the larger a game is, more is the complication of
the equilibrium strategies. Hence, such a player does not usually strategise
for the entire game right at the beginning, but does so incrementally. Af-
ter playing the game for a while, she observes the strategies that the other
players employed and the outcomes they received in the process. Depending
on this and her own outcomes, the player might then switch to a different
strategy in anticipation of countering the other players’ strategies and also
to attain better outcomes. The other players, on their part, of course ob-
serve this fact and hence they may themselves switch their own strategies.
Hence strategical reasoning by players in games is naturally mutually re-
cursive which results in their switching between ‘less complicated’ (atomic)
strategies to build (usually) ‘more complicated’ strategies. Thus switching
strategies constitutes a central aspect of strategic reasoning in games in
general and in large games in particular.

Strategies are traditionally defined to be functions from the set of histo-
ries of a game to the set of possible actions. In this sense, since combining
strategies by switching between them again results in such a function, the
resulting object is again a strategy. But this new strategy, instead of being
any arbitrary function, has well defined structure. Moreover, when the rules
of composition are given in some natural logical form, the structure of these
combined strategies is regular. Hence the long term effects of such strategies
on the game can be analysed and their outcomes predicted.

In this thesis, we start out by introducing the notion of strategy-switches
in traditional games: finite extensive form games, games on graphs as well as
repeated strategic form games. We observe that there are many interesting
questions one can ask about these games when the number of strategies used
to play the games and/or the number of switches between these strategies
is considered as a resource.

We first study the traditional two-player finite extensive form games as
well as games on graphs, with parity and Muller conditions where the strategies of players are restricted to a finite subset $S$ of the set of all strategies. We ask the following questions:

- Given a finite set of strategies $S$, is it possible for a player to play a winning strategy by just switching between the strategies in $S$?

- If so what is the minimum number of strategies / strategy-switches required from the set $S$?

We give algorithms for these questions and analyse their running time complexity.

Such questions are relevant in situations when playing a strategy involves a cost, maybe the cost of setting up the infrastructure required to play the strategy. We look at repeated strategic form games where the players incur a cost if they switch their strategies between two successive rounds. Under such a situation, would a player switch her strategy? Yes, if switching her strategy fetches her enough dividends in the long run to nullify these costs. However, a player who has been playing a certain strategy for a long time may settle down to a kind of an inertia or an unwillingness to switch her strategy even though she knows that doing so might fetch her better outcomes. For instance, although I know that changing my present house (with the leaking faucet and the creaking doors) to a different (newer) one would be far more comfortable for me, the very thought of shifting all the luggage and the furniture and disrupting my day-to-day schedule deters me from doing so. Games where switching strategies involve such costs have been studied in the literature (see for instance [LW97, LW09, Cha90] and the references therein). We re-prove some of the results in our setting to gain insight into what analytical changes these costs bring about.

We then turn our attention to large games. In this setting, as we observed already, there are various departures from the traditional setting of games: players may be uncertain about the number of players playing the game, the outcomes of the different rounds of the game and so on. In such games, it is natural to consider players with a bounded amount of resources: memory, computational power, knowhow, expertise, experience etc. Moreover, in such games the traditional way of studying equilibria can not only be unwieldy but also cannot be properly motivated. Hence, our analysis of such games is different from the way games have been traditionally analysed; we look at such games from an orthogonal point of view. More precisely, instead of studying the existence of winning/optimal strategies, we study
games where the strategies of the players are pre-specified (logically, algorithmically or by finite state automata). The question we ask of our model is the classical question of game theory: the prediction of stable behaviour in the limit. But what do we mean by stability in the context we consider?

In the case of players with bounded resources who are uncertain about the game, it is the short term changes that are under the players’ control. They switch between different strategies based on the outcomes of the play they observe. We are interested in questions concerning the eventual dynamics of the game:

- Which strategies survive eventually?
- What is the eventual outcome?
- Is stability with respect to switching attained?
- How worse-off are players employing heuristic strategies rather than playing best responses?

We study the above questions in a model where the strategies of the players have a compositional structure and are specified in terms of a simple syntax modelled on temporal and dynamic logics. We look at strategy-switching both from a logical and an algorithmic perspective.

In social situations, a natural consequence of switching strategies by players is that the game form itself changes intrinsically. The actions that are played by a small number of players may become too costly for the society to sustain and hence may cease to be available. This results in a change in the game form and we are interested in predicting which game forms finally emerge and remain stable given that the players play according to their strategy specifications and the society restricts the actions based on certain rules.

In this context, the converse question is also quite relevant and interesting: which actions of the players should the society restrict and how should it restrict them so that certain social goals are eventually achieved while minimising the cost? We address this question both in the case when the players are maximisers and when they play according to heuristics and show that in both cases it is enough for the society to keep track of a finite amount of information to generate the required action-restriction rules.

Our focus then shifts to the study of the rationale behind the strategy-switching by the players. As we already mentioned, in large games, resource-bounded players do not strategise for the entire game but do so dynamically revising their strategies based on certain heuristics. An important heuristic
Chapter 1. Games, logic and automata

to consider is that of imitation. Players tend to imitate other players with
better expertise and knowhow and of whom they know have performed well
in the game so far. As Dutta and Prasad [DP04] put it, “It is human to
imitate,” it is good for resource-bounded players to imitate as well! In this
context we study games where there are two types of players, ‘imitators’
and ‘optimisers’, and where each type is specified by a finite automaton.
We study the eventual outcomes in such games and show that one can give
a reasonable answer as to how worse-off the players are playing imitative
strategies rather than best responses. Imitation in games has been widely
studied, for instance in [Ban92, EF95, Sch98, LP07, DP04]. We, however,
take a different (automata theoretic) approach to analyse the effect of imi-
tation in large games.

For games involving a large number of players, we consider a model
where the players are arranged in certain neighbourhood structures. Such
a structure is given in the form of a (finite) graph where the vertices of the
graph represent the players and the edges represent the visibility relation
of the players. Similar models have been considered in the literature by var-
ious authors. Young [PY00, PY93] considers models where the interaction
structure of the players is represented by a finite undirected graph and stud-
ies how innovations spread through society by observation and interaction.
Kearns et al [KLS01a, KLS01b, EGG06, EGG07] analyse games where the
payoff of players depend only on her own action and the actions of her adja-
cent players as given by the graph structure. They study the computational
aspects and show the existence of Nash equilibria in such games.

In our setting a neighbourhood is a maximal clique in the neighour-
hood graph. Although the payoffs of the players are dependent only on the
actions of the other players in the same neighbourhood, their strategising
might depend on the actions and outcomes of all the players within their
visibility range, and in particular players outside their own neighbourhood.
Such a setting automatically provides a player with a rationale for switching
strategies. If she can observe a player in a different neighbourhood who is
performing better by playing a different strategy, she might switch to this
strategy and apply it in her own neighbourhood in the hope of doing bet-
ter. A player might even quit her own neighbourhood and join another one
expecting her strategy to fetch her better dividends in this new neighbour-
hood.

In this context we study games where the players play simple imitative
strategies and look at which actions and neighbourhood structures eventu-
ally arise and remain stable. We also characterise stable structures by relat-
ing them to potential games a la Monderer and Shapley [MS96]. Imitation
1.1. Organisation of the thesis

Dynamics in games have been studied, for instance, in [ARV06, AFBH08], where they study asymptotic time complexities of the convergence or non-convergence to Nash equilibrium in congestion games when the players play imitative strategies. However, their models are probabilistic and lack the neighbourhood structure that we consider.

In statistical physics, when dealing with large systems one uses averages or other aggregate measures and uses such measures to reason about the systems. In our case we work with anonymous utilities when the number of players are large. Such games have been called ‘anonymous games’ in the literature [Blo99, Blo00, DP07, BFH09]; anonymous because the utility of a player playing a certain strategy depends only on the number of other players playing the same strategy rather than the identity of the players. Such utilities can be thought of as a kind of an aggregate measure providing us with a handle on the system and simplifying the analysis.

When the number of players is large, the analysis is also simplified if we can obtain results on a smaller ‘derived’ system and then lift these results to the whole system. The concept of ‘types’ comes to our rescue in this regard. In large games, albeit there are a large number of players, but the number of different ways these players play is not so many. Each such way of play is called a ‘type’ and a player playing in such a manner is said to be of that type. For example, an ‘optimiser’ is a type and so also is an ‘imitator’ and so on. To this end, we show how and when the analyses of games with a large number of players can be carried out with a comparatively smaller number of player ‘types’ and still the results obtained can be applied to the entire game.

1.1 Organisation of the thesis

We present the thesis in two parts. Though each part can be considered more or less independent of the other, the central theme in both the parts is strategy switching by players. In Part-I we introduce strategy switching in games and in Part-II we mainly look at the structure of such strategy switches. We heavily use techniques from automata theory in both these parts and it can in a way be seen as bridging the gap between the two.

Part-I: Introducing strategy switching in games

Part-I consists of chapters 2, 3 and 4. In Chapter 2 we consider what happens when in a game the players are restricted to use strategies only from a fixed finite set $S$. We are interested in knowing if it is possible for a player to play
Chapter 1. Games, logic and automata

Winning / optimal strategies just by switching between strategies of \(S\) and if so how. We look at finite extensive form games, parity and Muller games in this context and study the computational complexity of these questions.

In Chapter 3 we study the case when switching strategies comes with a cost. We look at infinitely repeated strategic form games. A player incurs a fixed cost \(c\) if she switches her strategy from any round \(t\) to round \(t + 1\). Such games with switching costs have been studied in the literature. We re-prove some of the results for the sake of completeness.

In Chapter 4 we look at the eventual dynamics of concurrent-move games when the strategies of the players are explicitly specified. We introduce a logical syntax for the specification of the strategies of the players for this purpose. After that, we introduce the notion of probabilistic switching, which naturally generates mixed strategies. We again analyse the eventual outcome of the game when the players switch strategies probabilistically.

Part-II: Structure of strategy switching

Part-II consists of chapters 5, 6 and 7. In Chapter 5 we look at games that change intrinsically based on the actions / strategies played by the players. There is an implicit player - the society, who maintains the available actions of the players and incurs certain costs in making them available. If and when it feels that an action \(a\) is being played by a small number of players and/or it becomes too expensive for it to maintain the action \(a\), it removes \(a\) from the set of available actions. This results in a change in the game and the players have to strategise afresh taking this change into account. The restrictions of the society are again specified using a logical syntax. We are interested in which game forms eventually arise and remain stable thereafter.

We then study the converse question: which actions of the players should the society restrict and how should it restrict them so that the social cost is minimised in the eventuality? We address this question both in the case when the players are maximisers and when they play according to strategy specifications.

In Chapter 6 we look at imitation as a heuristic strategy for players. We consider \(n\)-player games where some of the players are optimisers (play best response strategies) and the rest are imitators. The players have preferences over the strongly connected components in the arena. We justify and consider the case where the strategies of both the imitators and the optimisers are given in terms of finite automata. We analyse how worse-off the players are playing imitative strategies rather than best-responses.
In Chapter 7 we look at games where the players are arranged in neighbourhoods. The neighbourhood structure is given by a graph $G$ which we call the neighbourhood graph. The vertices of the graph correspond to the players and the cliques in $G$ are the neighbourhoods and the edges in $G$ correspond to the visibility structure of the players. Although the payoffs of the players are affected only by the moves of the players in her own neighbourhoods, their strategising can depend on what they can view of their own and also other neighbourhoods. We consider two cases, one where the players stick to their own neighbourhoods throughout the course of the game and the other where players can also switch neighbourhoods. We study games with both of the above neighbourhood structures where the players play simple imitative strategies. We also study general games with such neighbourhood structures.

Finally, in Chapter 8 we conclude with future directions and possible extensions.

1.2 Background

In this section we develop the necessary preliminaries for the rest of the thesis.

1.2.1 Strategic form game

A strategic form or normal form game consists of a set $N = \{1, 2, \ldots, n\}$ of players and a collection of moves (strategies) $A_i$ for each player $i$. The game is played in a single round in which each player $i$ chooses a move $a_i$ from $A_i$. This is done simultaneously and a player $i$ does not know the move chosen by any other player $j$ before she makes her choice. This defines a tuple $a = (a_1, a_2, \ldots, a_n) \in A = \prod_{i \in N} A_i$. Each player $i$ has a preference $\preceq_i$ over the various tuples in $A$. When this preference is a total order, we can represent it in the form of a payoff function $p_i : A \to \mathbb{N}^1$. Thus a strategic form game between $n$ players is an $n$-dimensional matrix where each dimension of the matrix corresponds to a player $i$, the indices of the dimension correspond to the moves available to the player $i$ and the entries of the matrix correspond to the payoff tuples of the players with respect to the indices.\footnote{As we consider only ordinal preferences as opposed to expected utilities, such a representation is without loss of generalisation}
1.2.2 Extensive form game arena

Throughout this exposition, we shall investigate the long-run dynamics of games, viz., eventual behaviour of players, eventual outcomes etc. Local or small temporal perturbations should not affect the global outcome or the eventual dynamics of the games. Hence, in such a setting, it is natural to assume that the play is ‘unbounded’, or in other words continues for an infinite amount of time. This has the advantage that small perturbations and changes, ‘losses’ and ‘profits’, mistakes and risks etc. are amortised away and need not affect eventual outcomes very much. Thus, although we shall look at finite games, our focus shall mostly be on infinite duration games.

In the above spirit, we define an extensive form arena to be a (finite or infinite) tree $T = (T, E)$ where $T$ is the set of vertices and $E$ is the set of edges. The root of the tree $t_0 \in T$ is the initial vertex of the arena. We call the pair $(T, t_0)$ an initialised arena. For a vertex $t \in T$, we let $E(t)$ denote the set of outgoing edges of $t$ and $tE$ denote the set of children of $t$. That is, $E(t) = \{(t, t') \in E\}$ and $tE = \{t' \mid (t, t') \in E\}$. $N = \{1, 2, \ldots, n\}$ is the set of players and $A_i$ is the set of actions of player $i \in N$. Let $A = \prod_{i \in N} A_i$ and $A = \bigcup_{i \in N} A_i$. For every player $i$, there is a function $\Gamma_i : T \to 2^{A_i}$ such that $\Gamma_i(t)$ gives the set of actions that are available to the player $i$ at the vertex $t$.

The edges of the arena $T$ are labelled with tuples from $A$. For simplicity, we assume that for every vertex $t \in T$ and for every tuple of actions $a \in \prod_{i \in N} \Gamma_i(t)$, there is an edge $(t, t')$ that is labelled with $a$. A game is played on this arena as follows. Initially a token is placed at the root $t_0$ of $T$. Whenever the token is at some vertex $t$, every player $i$ chooses an action $a_i$ from her set of available actions $\Gamma_i(t)$. This defines a tuple $a = (a_1, a_2, \ldots, a_n) \in A$. The token then moves along the outgoing edge of $t$ labelled with $a$ to the corresponding child. This process defines a branch of the tree $T$ which is called a play. We denote the set of plays by $P$. A finite play, which is a prefix of a branch of $T$, is called a history. We denote the set of all histories by $H$.

**Example 1.1** Consider an extensive form arena where there are two players 1 and 2 and at each position player 1 has two actions $\{a, b\}$ and player 2 has two actions $\{c, d\}$. This arena looks like Figure 1.1. We have simply mentioned the choices that are available at each node and have avoided writing down the entire sequence so as not to clutter the figure. For instance, the leftmost grandchild of the root node is actually $aa, cc$ and not $a, c$. However, this is clear from the context.
1.2. Background

A turn-based arena is a special case of the above arenas where the vertex set \( T \) is partitioned into \( T = T_1 \cup T_2 \cup \ldots \cup T_n \) with the restriction that for any player \( i \) and any vertex \( t \in T_i, \Gamma_j(t) \) is a singleton for all \( j \neq i \). Thus, a turn-based arena is one where the players take turns in making moves in that at a vertex \( t \in T_i \) it is the player \( i \)'s choice that determines the next vertex; the other players do not have any control over it. We generally write a play \( \rho = t_0 a_1 t_1 a_2 \ldots \) as \( \rho = t_0 a_1 t_1 a_2 \ldots \) where \( a_1 = a_1(i) \) such that \( t_0 \in T_i \), \( a_2 = a_2(j) \) such that \( t_1 \in T_j \) and so on. Since at a vertex \( t \), only one player has a non-trivial choice, such a representation is without loss of generality.

1.2.3 Graph arena

We wish to do algorithmic analysis on the extensive form arenas. For this we require that these arenas be presented to us in a finite fashion (when they are infinite). A natural way to present a game arena is by a set of rules, stating which moves are possible at what position and what will be the resulting position. See [KS10] for a treatment along these lines. We however take a more direct approach and let our extensive form arenas be unfoldings of finite directed graphs.

A graph arena is thus a directed graph \( \mathcal{A} = (V, E) \) where \( V \) is the set of vertices and \( E \) is the set of edges. As in the case of extensive form arenas, for a vertex \( v \in V \), we denote by \( vE \) the set of all the neighbours of \( v \) and by \( E(v) \) the set of all outgoing edges of \( v \). For simplicity we assume that \( \mathcal{A} \) has no dead-ends, that is, \( vE \neq \emptyset \) for all \( v \in V \). As before, the set of players is \( N = \{1, 2, \ldots, n\} \). An initialised arena \( (\mathcal{A}, v_0) \) is one where an vertex \( v_0 \in V \) is designated as an initial vertex. A subarena of \( \mathcal{A} \) is a subgraph of \( \mathcal{A} \) with no dead-ends.
Every player $i$ has a finite set of actions $A_i$. Let $A = \prod_{i \in \mathbb{N}} A_i$ and $A = \bigcup_{i \in \mathbb{N}} A_i$. For every player $i$, there is a function $\Gamma_i : V \to 2^{A_i}$ such that $\Gamma_i(v)$ gives the set of actions that are available to the player $i$ at the vertex $v$. The edges of the arena $A$ are labelled with tuples from $A$. We once more assume that for every vertex $v \in V$ and for every tuple of actions $a \in \prod_{i \in \mathbb{N}} \Gamma_i(v)$, there is an edge $(t, t')$ that is labelled with $a$. A game is played on this arena as follows. Initially a token is placed at the initial vertex $v_0$ of $V$. Whenever the token is at some vertex $v$, every player $i$ chooses an action $a_i$ from her set of available actions $\Gamma_i(v)$. This defines a tuple $a = (a_1, a_2, ..., a_n) \in A$. The token then moves along the outgoing edge of $v$ labelled with $a$ to the corresponding neighbour. This process defines a path in $A$ which is called a play. We denote the set of plays by $P$. A finite play is called a history. We denote the set of all histories by $H$.

Since we have assumed that our arenas have no dead-ends, all plays in such an arena are always infinite.

**Turn-based arena**

A turn-based arena is again a special case of the above arenas where the vertex set $V$ is partitioned into $V = V_1 \cup V_2 \cup \ldots \cup V_n$ with the restriction that for any player $i$ and any vertex $v \in V_i$, $\Gamma_j(v)$ is a singleton for all $j \neq i$.

Once more, without loss of generality, we write a play $\rho = v_0 a_1 \rightarrow v_1 a_2 \rightarrow \ldots$ as $\rho = v_0 \xrightarrow{a_1} v_1 \xrightarrow{a_2} \ldots$ where $a_1 = a_1(i)$ such that $v_0 \in V_i$, $a_2 = a_2(j)$ such that $v_1 \in V_j$ and so on.

We would like to talk about various properties of a game position in logical terms. For this purpose we have a set of propositions $\mathcal{P}$ and a valuation function

\[ \text{val} : V \to 2^\mathcal{P} \]

which gives the truth of these propositions at the vertices of the arena. These propositions can stand for facts of the form “player $i$ received the highest payoff in round $t$”, “player $i$ played action $a$ in round $t$” etc. Intuitively, these propositions are observations about the game based on which the players strategise.

**1.2.4 Some notations**

For a directed graph $G = (V, E)$ where the edges are labelled with elements from a set $X$, if $(v, v')$ is an edge in $E$ labelled with $x$ then we often denote it by $v \xrightarrow{x} v'$. For a finite path $\rho = v_0 \xrightarrow{x_1} v_1 \xrightarrow{x_2} \ldots \xrightarrow{x_k} v_k$ in $G$ we let $\text{last}(\rho) = v_k$. For a finite or infinite path $\rho = v_0 \xrightarrow{x_1} v_1 \xrightarrow{x_2} \ldots$, we let
For a finite sequence \( u = x_1x_2 \ldots x_k \in X^* \) and given a vertex \( v \in V \), we let \( \rho(v, u) \) be the path \( v \xrightarrow{a_1} v_1 \xrightarrow{a_2} v_2 \xrightarrow{a_3} \ldots \xrightarrow{a_k} v_k \) in \( G \). For a set \( X \) and for a sequence \( u = x_0x_1 \ldots x_k \in X^* \), we similarly let \( \text{last}(u) = x_k \). For a sequence \( u = x_1x_2 \ldots \in X^* \), we let \( u(i) \) be the \( i \)-th element of \( u \), \( u_i \) be the length \( i \) prefix of \( u \), and \( u^i \) be the postfix of \( u \) starting at the \( i \)-th element, that is, \( u^i = x_ix_{i+1} \ldots \), where \( i < |u| \), the length of the sequence \( u \). For \( x \in X \) and a sequence \( u \in X^* \), we let \( |u|_x \) be the number of occurrence’s of \( x \) in \( u \). We let \( \text{occ}(u) \) be the set of elements that occur at least once in \( u \), that is, \( \text{occ}(u) = \{ x \mid \exists i, \ u(i) = x \} \). The above notations generalise naturally to the set of infinite sequences over \( X \), \( X^\omega \).

For \( u \in X^\omega \), we let \( \text{inf}(u) \) be the set of elements that occur infinitely often in \( u \), that is, \( \text{inf}(u) = \{ x \mid \forall i \exists j > i, \ u(j) = x \} \). For a tuple \( a \in A \), we let \( a(i) \) be the \( i \)-th component of \( a \).

### 1.2.5 Unfolding

The tree-unfolding or just the unfolding of an initialised graph arena \((A, v_0)\) where \( A = (V, E) \) is a tree \( \mathcal{T}_A = (T, E) \) such that \( T = V \times A^* \) and is defined inductively as follows:

- \( t_0 = (v_0, \epsilon) \in T \) is the root of \( \mathcal{T}_A \).

- For every \( t \in T \), the children of \( t \) are derived as follows. If \( t = (v, u) \) then for every \( a \in A \), \((v', ua)\) is a child of \( t \) where \( v' \in vE \) and the edge \((v, v')\) is labelled with \( a \) in \( A \).

\( \mathcal{T}_A \) is the extensive form game arena corresponding to the graph arena \( A \). The valuation of the propositions \( P \) on \( V \) is lifted naturally to \( T \) as: for every \( t \in T \), \( \text{val}(t) = \text{val}(v) \) where \( t = (v, u) \). For a path \( \rho = t_1 \xrightarrow{a_1} t_2 \xrightarrow{a_2} \ldots \xrightarrow{a_{k-1}} t_k = (v_1, u_1) \xrightarrow{a_1} (v_2, u_2) \xrightarrow{a_2} \ldots \xrightarrow{a_{k-1}} (v_k, u_k) \) in the unfolding \( \mathcal{T}_A \), we let \( \pi(\rho) \) denote the projection of \( \rho \) to its first component, that is, \( \pi(\rho) = v_1 \xrightarrow{a_1} v_2 \xrightarrow{a_2} \ldots \xrightarrow{a_{k-1}} v_k \).

**Example 1.2** Let \( N = \{1, 2\} \) be the players, \( A_1 = \{a, b\} \) be the action set of player 1 and \( A_2 = \{c, d\} \) be the action set of player 2. Consider the arena \( A \) shown in Figure 1.2 consisting of the two positions \( v_0 \) and \( v_1 \) where \( v_0 \) is the initial position. The tree unfolding of the above arena is the tree depicted in Example 1.1.
Chapter 1. Games, logic and automata

![Diagram of a game arena presented as a finite graph](image)

**Figure 1.2: A game arena presented as a finite graph**

### 1.2.6 Winning condition

An $n$-player game typically consists of an arena $\mathcal{A}$ and winning conditions $\phi_1, \phi_2, \ldots, \phi_n$. The winning conditions are subsets of the set of all plays in the arena, that is, $\phi_i \subseteq P$ for $1 \leq i \leq n$. The objective of every player $i$ is to play so that the play is in her winning set $\phi_i$. Such games are also called win-lose games because of the obvious reason that a player can either win or lose. However, note that the winning sets may not be disjoint, that is, it may be that $\phi_i \cap \phi_j \neq \emptyset$ for some $i \neq j$. If the winning sets form a partition of the set of plays $P$ then the game is called zero-sum. Two-player zero-sum games have a single winning set $\phi$ with the convention that player 1 wins a play $\rho$ in the arena $\mathcal{A}$ if and only if $\rho \in \phi$. Otherwise it is losing for her and winning for player 2. Various types of winning conditions have been studied in the literature for the case of two-player zero-sum games.

**Borel condition:** $\phi$ is a Borel set in the Cantor topology on the set of plays in the arena $\mathcal{A}$.

**$\omega$-regular condition:** this is a general sub-class of the Borel winning condition and can be presented in different ways. $\omega$-regular winning conditions naturally arise as sets specified by various specification languages. Three commonly used $\omega$-regular conditions can be described as follows:

**Reachability condition:** Let $R \subseteq V$, $R \neq \emptyset$ be the reachability set. A play $\rho$ in the arena $\mathcal{A}$ is said to be winning, $\rho \in \phi$, if and only if

$$\exists i \geq 0, \rho(i) \in R.$$ 

Let $C \subset \mathbb{N}$ be a finite set of colours also called priorities. Let $\chi$ be a function that assigns a unique priority to each of the vertices in $V$:

$$\chi : V \to C$$
χ is lifted to sequences in $V^*$ or $V^\omega$ as: for $u = v_1v_2\ldots$, $\chi(u) = \chi(v_1)\chi(v_2)\ldots$. χ can also be lifted to plays in $\mathcal{A}$ as: let $\rho = v_0 \xrightarrow{a_1} v_1 \xrightarrow{a_2} v_2 \xrightarrow{a_3} \ldots$ be a play in $\mathcal{A}$. Then $\chi(\rho) = \chi(v_0v_1v_2\ldots)$.

**Muller condition:** Let $F \subseteq 2^C$ be a family of subsets of $C$ called the Muller sets. A play $\rho$ in the arena $\mathcal{A}$ is said to be winning if and only if

$$\inf(\chi(\rho)) \in F.$$ 

**Parity condition:** A play $\rho$ in the arena $\mathcal{A}$ is said to be winning, $\rho \in \phi$, if and only if

$$\min\{\inf(\chi(\rho))\}$$ is even.

**Mean-payoff condition:** is also a sub-class of the Borel condition. Here the vertices of the arena are labelled with weights or rewards. That is, there is a function $r : V \to \mathbb{Q}$ which associates a rational number with every vertex of the arena. A play $\rho$ is winning, $\rho \in \phi$, if and only if the following quantity

$$\lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k} r(\rho(i))$$

is greater than a certain threshold, usually 0.

Another variation of $n$ player games studied in the literature is where every player $i$, instead of having a winning set $\phi_i$, has a preference relation $\sqsubseteq_i$ over the various subsets of $C$ (the Muller sets). One of the aims of the players is to play in such a way that no player has an incentive to unilaterally deviate from such a play $\rho$. Player $i$ then receives a payoff that is proportional to her preference of the play $\rho$ in terms of her preference relation $\sqsubseteq_i$. Such a plan is called a Nash equilibrium and is defined in the next section.

### 1.2.7 Strategy

A strategy of a player tells her how to play the game. Formally, a strategy $s_i$ of player $i$ is a function

$$s_i : H \to A_i$$

A play $\rho$ in the arena $\mathcal{A}$ is said to conform to a strategy $s_i$ if for all $k \geq 0$ and all length $k$ and $k+1$ prefixes $\rho_k$ and $\rho_{k+1}$ respectively of $\rho$, $\text{last}(\rho_k) \xrightarrow{a} \text{last}(\rho_{k+1})$ implies $a(i) = s_i(\rho_k)$. 

29
Strategies can also be randomised. In that case, the strategy of a player at a particular history does not prescribe her one action but a probability distribution over the set of available actions. Such strategies are called mixed strategies. Formally, let $\Delta(A_i)$ be the set of probability distributions over the set $A_i$ of actions of player $i$. Then, a mixed strategy $s_i$ of player $i$ is a function

$$s_i : H \to \Delta(A_i)$$

A non-randomised strategy $s_i$ of a player $i$ is then a special case of a mixed strategy where for every history $\rho$, $s_i(\rho)$ has positive support on exactly one action of $A_i$. These strategies are also called pure strategies. Henceforth, we shall use the word ‘strategy’ to denote pure strategies and when we want to talk about mixed strategies, we shall say so explicitly.

Apart from the above definitions of a strategy, we shall also be interested in partial strategies where a strategy of any player $i$ is a partial function from the set of histories to the set of actions $A_i$ of player $i$. We adopt the convention that if a partial strategy $s_i$ of a player $i$ is not defined at a particular history $\rho$ then she may play any action there. Thus a play $\rho$ in the arena $\mathcal{A}$ is said to conform to strategy $s_i$ if for all $k \geq 0$ and all length $k$ and $k+1$ prefixes $\rho_k$ and $\rho_{k+1}$ respectively of $\rho$, $\text{last}(\rho_k) \xrightarrow{a} \text{last}(\rho_{k+1})$ and $s_i$ is defined on $\rho_k$ implies $a(i) = s_i(\rho_k)$.

For every player $i \in N$ we let $\Sigma_i$ denote the set of all strategies of $i$. A strategy profile $(s_1, \ldots, s_n)$ is a subset of $\Sigma_1 \times \ldots \times \Sigma_n$.

**Strategy tree**

A strategy $s_i$ of a player $i$ in an arena $\mathcal{A}$ (turn-based or concurrent) can be viewed as a subtree of the extensive form arena $T_\mathcal{A}$ corresponding to $\mathcal{A}$. We call this subtree the strategy tree of $s_i$ and denote it as $T_{s_i}^{\mathcal{A}}$. $T_{s_i}^{\mathcal{A}}$ is obtained from $T_\mathcal{A}$ by retaining only the plays that conform to $s_i$. Formally $T_{s_i}^{\mathcal{A}} = (T^{s_i}, E^{s_i})$ such that

- $t_0 = (v_0, \epsilon) \in T^{s_i}$ is the root of $T_{s_i}^{\mathcal{A}}$.
- For every vertex $t = (v, u)$ in $T^{s_i}$, let $\rho$ be the path from $t_0$ to $t$ and let $\pi(\rho)$ be the projection of this path to the first component. If $s_i(\pi(\rho)) = a$ then the children of $t$ are all $t'$ such that $t' = (v', u)u$ where $v \xrightarrow{a} v'$ in $\mathcal{A}$ and $a(i) = a$.
- Nothing else is in $T_{s_i}^{\mathcal{A}}$. 


1.2. Background

Example 1.3 In the extensive form game of Example 1.1, let \( s_1 \) be the strategy of player 1 which prescribes her to play the action \( a \) at every history. Then the strategy tree \( T_{A_1}^s \) looks like Figure 1.3, which is a subtree of \( T_A \).

Example 1.4 In the extensive form game of Example 1.1, let \( s'_1 \) be the partial strategy for player 1 which is undefined at the empty history but prescribes her to play the action \( a \) for all successive histories. Then \( T_{A_1}^{s'_1} \) looks as shown in Figure 1.4.

Determinacy

Let \( A = (V, E) \) be a turn-based arena and let \( \phi_i \) be the winning condition for player \( i \). A strategy \( s_i \) for player \( i \) is called a winning strategy at a vertex \( v \in V \) if for every play \( \rho \) that starts at \( v \) and conforms to \( s_i \) is winning for \( i \), that is, we have \( \rho \in \phi_i \). The biggest subset \( W_i \subseteq V \) such that player \( i \) has a winning strategy at every vertex \( v \in W_i \) is called the winning region for \( i \). A two-player zero-sum turn-based game \( (A, \phi) \) (\( \phi \) is the winning condition of player 0) is said to be (qualitatively) determined if for every vertex \( v \in V \)
it is the case that either player 0 or player 1 has a winning strategy at $v$. A classical problem in the theory of infinite two-player zero-sum games on graphical arenas is to determine given a game $(A, \phi)$, if it is determined and if so to compute the winning regions and the winning strategies of both the players.

For concurrent games, such (pure-strategy) determinacy does not hold. For such games, it can be shown that randomised strategies are more powerful than pure strategies. Let $A = (V, E)$ be a two-player zero-sum concurrent arena and let $\phi$ be the winning condition of player zero. Given a vertex $v \in V$, the maximum probability with which player 0 can ensure $\phi$ from $v$ is called the value of the game at $v$ for player 0 and is denoted $val_0(\phi)(v)$. Similarly the maximum probability with which player 1 can ensure $\bar{\phi} = P \setminus \phi$ from $v$ is called the value of the game at $v$ for player 1 and is denoted $val_1(\phi)(v)$. A concurrent two-player zero-sum game $(A, \phi)$ is said to be (quantitatively) determined if for every vertex $v \in V$ it is the case that $val_0(\phi)(v) + val_1(\phi)(v) = 1$.

Equilibrium

In the non zero-sum setting, determinacy or winning regions cannot be defined. Various solution concepts have been explored in such a setting. Nash equilibrium is perhaps the most widely studied of these. Let $A = (V, E)$ be an arena. Suppose the players have individual preferences over the various plays in this arena and suppose the preference relation of player $i$ is denoted by $\sqsubseteq_i$. Let $s = (s_1, s_2, \ldots, s_n)$ be a strategy profile of the players. We let $s_{-i}$ be the profile $s$ with the $i$th component removed and $(s_{-i}, s_i)$ be the profile $s$ except that the $i$th component is the strategy $s_i$. Given a strategy profile $s$, we let $\rho(s)$ be the unique play in the arena conforming to $s$.

A strategy $s$ of player $i$ is called a best response to a strategy profile $s_{-i}$ of the other players if for every strategy $s'_i$ of player $i$, $\rho(s_{-i}, s'_i) \sqsubseteq_i \rho(s_{-i}, s)$. A strategy profile $s$ is said to be a Nash equilibrium if for every $i$, $s_i$ is a best response to $s_{-i}$.

Finite memory strategy

A strategy $s$ of player $i$ is said to be finite memory if it can be presented as a tuple $(M, m^I, \delta, g)$ where $M$ is a finite set called the memory of the strategy, $m^I \in M$ is called the initial memory, $\delta : A \times M \to M$ is called the memory update function, and $g : A \times M \to A_i$ is called the output function such that if $a_0a_1 \ldots a_k$ is a play and $m_0m_1 \ldots m_{k+1}$ is a sequence determined by
1.2. Background

$m_0 = m^I$ and $m_{j+1} = \delta(a_j, m_j)$ then $s(a_0 a_1 \ldots a_k) = g(a_k, m_{k+1})$. The strategy $s$ is said to be memoryless or positional if $M$ is a singleton. Finite memory strategies can be modelled using finite state transducers.

1.2.8 Finite state transducer

A finite state transducer (FST) over input alphabet $X$ and output alphabet $Y$ is a tuple $Q = (Q, I, \delta, f)$ where

- $Q$ is the set of states,
- $I \subseteq Q$ is the set of initial states,
- $\delta : Q \times X \rightarrow 2^Q$ is the transition function and
- $f : Q \rightarrow Y$ is the output function.

A finite memory strategy, as defined above, can thus be naturally represented in terms of a finite state transducer. Let $s_i = (M, m^I, \delta, g)$ be a finite

memory strategy for player $i$. $s_i$ is represented by an FST $Q_{s_i} = (Q, I, \delta, f)$ over input alphabet $A$ and output alphabet $A_i$ where the states $Q$ of $Q_{s_i}$ is equal to the memory $M$ of the strategy $s_i$, the transition relation $\delta$ of $Q_{s_i}$ is the same as the memory update function $\delta$ of $s_i$, the set of initial states $I$ is equal to the singleton $\{m^I\}$ and the output function $f$ is equal to $g$.

An FST $Q_{s_i} = (Q, I, \delta, f)$ corresponding to a finite memory strategy $s_i$ can be run on a subtree $T = (T, E)$ of an extensive form game tree (unfolding). A run $r$ of $Q_{s_i}$ on $T$ is a function $r : T \rightarrow Q$ that labels the vertices of $T$ with states of $Q_{s_i}$ and is defined inductively as:

- $r(t_0) \in I$.
- $r(t) = q$ and $t \xrightarrow{a} t' \in T$ implies $t' \in \delta(q, a)$.

A tree $T$ is said to be accepted by $Q_{s_i}$ if there exists a run $r$ of $Q_{s_i}$ on $T$ such that for every $t \xrightarrow{a} t'$, suppose $r(t) = q$, $t = (v, u)$ and suppose $\rho$ is the path from $t_0$ to $t$. If $\pi(\rho)$ is the projection of this path to its first component, we have $a(i) = s_i(\pi(\rho))$.

The language of $Q_{s_i}$ is defined to be the set $\mathcal{L}(Q_{s_i})$ of all the trees accepted by it. Note that the set $\mathcal{L}(Q_{s_i})$ is the set of all strategy trees of the strategy $s_i$.2

2This is a regular tree language in the parlance of automata theory
Chapter 1. Games, logic and automata

Product of transducers

Let $Q_1 = (Q_1, I_1, \delta_1, f_1)$ and $Q_2 = (Q_2, I_2, \delta_2, f_2)$ be two FSTs. The product of $Q_1$ and $Q_2$ is defined as $Q = Q_1 \times Q_2 = (Q, I, \delta, f)$ where

- $Q = Q_1 \times Q_2$
- $I = I_1 \times I_2$
- $\delta = \{((q_1, q_2), a, (q'_1, q'_2)) \mid (q_1, a, q'_1) \in \delta_1, (q_2, a, q'_2) \in \delta_2\}$
- $f: Q \rightarrow A_1$ or $f: Q \rightarrow A_2$ such that $f(q_1, q_2) = f_1(q_1)$ or $f(q_1, q_2) = f_2(q_2)$ depending on the operation at hand.

1.2.9 A modal logic

To reason about the properties of the various positions in an extensive form game, we develop a modal logic. The syntax of this logic is similar to temporal logic. However, its formulas are evaluated on nodes of extensive form game trees and hence they talk about a unique past and branching future. The syntax of this logic is formally given as:

$$\Phi ::= p \in P \mid \neg \varphi \mid \varphi_1 \lor \varphi_2 \mid \langle a \rangle \neg \varphi \mid \langle a \rangle^+ \varphi \mid \Diamond \varphi$$

We have a couple of fragments of this logic. First, when we do not wish to talk about the future we have the logic $\Phi^-$ which is derived from $\Phi$ by dropping the modality $\langle a \rangle^+ \varphi$. Similarly, when we do wish to talk about bounded future but not the past we have the logic $\Phi^+$ which is derived from $\Phi$ by dropping $\langle a \rangle \neg \varphi$ and $\Diamond \varphi$.

We also use the standard abbreviations: $\Box \varphi \equiv \neg \Diamond \neg \varphi$, $\Diamond \varphi \equiv \lor_{a \in A} \langle a \rangle^+ \varphi$ and $\lor \varphi \equiv \lor_{a \in A} \langle a \rangle^+ \varphi$.

A formula $\varphi \in \Phi$ is evaluated on the vertices of $T_A$. The truth of a formula $\varphi$ at a vertex $t = (v, u) \in T_A$ is denoted by $t \models \varphi$ and is defined inductively as follows.

- $t \models p$ iff $p \in val(v)$.
- $t \models \neg \varphi$ iff $t \not\models \varphi$.
- $t \models \varphi_1 \lor \varphi_2$ iff $t \models \varphi_1$ or $t \models \varphi_2$.
- $t \models \langle a \rangle \neg \varphi$ iff there exists $t'$ in $T_A$ such that $a \rightarrow t$ and $t' \models \varphi$.
- $t \models \langle a \rangle^+ \varphi$ iff there exists $t'$ in $T_A$ such that $a \rightarrow t'$ and $t' \models \varphi$.
- $t \models \Diamond \varphi$ iff there exists an ancestor $t'$ of $t$ such that $t' \models \varphi$.
Fischer-Ladner closure, atom and atom graph

For a formula $\varphi \in \Phi$ we define its Fischer-Ladner closure $CL(\varphi)$ as follows.

First we construct the subformula closure of $\varphi$, $CL'(\varphi)$ as:

- $\varphi \in CL'(\varphi)$.
- $\neg \varphi' \in CL'(\varphi)$ implies $\varphi' \in CL'(\varphi)$.
- $\varphi_1 \lor \varphi_2 \in CL'(\varphi)$ implies $\varphi_1 \in CL'(\varphi)$ and $\varphi_2 \in CL'(\varphi)$.
- $\langle a \rangle^- \varphi' \in CL'(\varphi)$ implies $\varphi' \in CL'(\varphi)$.
- $\langle a \rangle^+ \varphi' \in CL'(\varphi)$ implies $\varphi' \in CL'(\varphi)$.
- $\exists \varphi' \in CL'(\varphi)$ implies $\exists \varphi', \varphi' \in CL'(\varphi)$.

Finally let $CL(\varphi) = CL'(\varphi) \cup \{\neg \varphi' \mid \varphi' \in CL'(\varphi)\}$ where we identify $\neg \neg \varphi'$ with $\varphi'$.

A set $C \subseteq CL(\varphi)$ is called an atom (or a maximal propositionally consistent subset of $CL(\varphi)$) if:

- $\forall \varphi' \in CL(\varphi), \varphi' \in C$ iff $\neg \varphi' \notin C$.
- $\forall \varphi_1 \lor \varphi_2 \in CL(\varphi), \varphi_1 \lor \varphi_2 \in C$ iff $(\varphi_1 \in C \text{ or } \varphi_2 \in C)$.
- $\forall \varphi' \in CL(\varphi), \exists \varphi' \in C$ iff $(\varphi' \in C \text{ or } \exists \varphi' \in C)$.

We denote by $AT(\varphi)$ the set of all atoms of $\varphi$. For any atom $C \in AT(\varphi)$ let $val(C) = C \cap P$.

The atom graph $G_\varphi = (V_\varphi, E_\varphi)$ of a formula $\varphi \in \Phi$ is constructed as follows:

i. $V_\varphi = AT(\varphi)$.

ii. $E_\varphi \subseteq V_\varphi \times A \times V_\varphi$ such that $(C, a, C') \in E_\varphi$ if and only if for all $\langle a \rangle^- \varphi' \in CL'(\varphi), \langle a \rangle^- \varphi' \in C' \iff \varphi' \in C$.

iii. $(C, a, C') \in E_\varphi$ if and only if for all $(a)^+ \varphi' \in CL'(\varphi), (a)^+ \varphi' \in C \iff \varphi' \in C'$.

The atom graph for a formula $\varphi$ in the fragments $\Phi_-$ and $\Phi_+$ can be suitably defined. A vertex $C$ in the atom graph $G_\varphi$ is (usually) designated an initial node only if it does not have any past requirement, that is, it does not have any formula of the form $\langle a \rangle^- \varphi'$ or $\langle a \rangle^- \varphi'$. Hence note that for any $\exists \varphi' \in CL(\varphi), \exists \varphi' \in C$ implies $\varphi' \in C$. 
We call a subgraph $G'_{\varphi} = (V'_{\varphi}, E'_{\varphi})$ of $G_{\varphi}$ ‘good’ if for every $C \in V'_{\varphi}$, there exists $C' \in V'_{\varphi}$ such that $C$ is reachable from $C'$ and $C'$ is initial.

Let $\mathcal{T}_{G'_{\varphi}}$ be the unfolding of a good subgraph $G'_{\varphi}$ of the atom graph $G_{\varphi}$ starting at an initial vertex $C_0$ of $G'_{\varphi}$. We have the following proposition

**Proposition 1.5** For any node $t = (C, u)$ of $\mathcal{T}_{G'_{\varphi}}$ and for every $\alpha \in CL(\varphi)$ we have $t \models \alpha$ iff $\alpha \in C$.

**Proof** The proof is by induction on the structure of $\alpha$.

- $\alpha = p \in \mathcal{P}$: Follows from definition since $t \models \alpha$ iff $p \in \text{val}(C)$ iff $p \in C$.
- $\alpha = \neg \alpha'$: $t \models \neg \alpha$ iff $t \not\models \alpha'$ iff $\alpha' \notin C$ [since $C$ is an atom].
- $\alpha = \alpha_1 \lor \alpha_2$: $t \models \alpha$ iff $t \models \alpha_1$ or $t \models \alpha_2$ iff $\alpha_1 \in C$ or $\alpha_2 \in C$ [since $C$ is an atom].
- $\alpha = (\alpha)^\neg$: $t \models (\alpha)^\neg$ iff $t' \not\models \alpha$ where $t' = (C', u')$ iff $(\alpha)^\neg \in C$ [by the construction of the atom graph].
- $\alpha = (\alpha)^\dagger$: Similar to the case for $(\alpha)^\neg$.

For the converse direction, suppose that $\Diamond \alpha' \in C$. Since $G'_{\varphi}$ is a good subgraph, $C$ is reachable from the initial vertex $C_0$. Hence, by the definition of the atom graph there exists a vertex $C''$ on the path from $C_0$ to $C$ such that $\alpha' \in C''$. Let $C'$ be the last such vertex. We induct again on the distance between $C''$ and $C$. Let $d$ denote this distance. The base case is when $d = 0$. Then we have that $\alpha' \in C$ and hence $\Diamond \alpha' \in C$ [since $C$ is an atom] which implies $t \models \Diamond \alpha'$ [by definition]. Let $d > 0$ and let $t'' = (C'', u'')$ be the parent of $t$. Since the distance between $C'$ and $C''$ is $d - 1$ we can apply the second induction hypothesis to conclude that $t'' \models \Diamond \alpha'$. Hence $t \models \Diamond \alpha'$ [by definition].

$\square$
1.2.10 Markov chain

A Markov chain is a tuple $\mathcal{M} = (V, E, \delta)$ where $(V, E)$ is a directed graph and $\delta$ is a function $\delta : V \to \Delta(V)$ where $\Delta(V)$ is the set of probability distributions on $V$ such that for $v \in V$, $\delta(v) = p_v$ if and only if $v' \notin vE$ implies $p_v(v') = 0$. Thus for every vertex $v \in V$, $\delta(v)$ gives a probability distribution $p_v$ which gives the probabilities with which each outgoing edge of $v$ is selected. A Markov chain is called finite if $V$ is finite.

Given a Markov chain $\mathcal{M} = (V, E, \delta)$ and an initial vertex $v_0 \in V$ we can define a process on $\mathcal{M}$ as follows. Initially a token is placed at $v_0$. Whenever a token is in some vertex $v \in V$, a coin is tossed and the token is moved to a neighbour $v' \in vE$ with the probability $\delta(v)(v')$. If $(V, E)$ has no dead-ends, then this process goes on forever.

Let $\mathcal{M} = (V, E, \delta)$ be a finite Markov chain and let the vertices $V$ be ordered as $V = \{v_1, v_2, \ldots, v_{|V|}\}$. We can associate an $|V| \times |V|$ matrix $M$ with $\mathcal{M}$ such that the $ij$ th entry $m_{ij}$ of $M$ is the probability of going from the vertex $v_i$ to $v_j$ in the Markov chain $\mathcal{M}$, that is, $m_{ij} = \delta(i)(j)$. Such a matrix $M$ is called the transition matrix of $\mathcal{M}$.

A vector $q = (q_1, q_2, \ldots, q_{|V|})$ such that $\sum_{i=1}^{|V|} q_i = 1$ and $q_i \geq 0$ for all $i : 1 \leq i \leq |V|$ is said to be a stationary distribution of a Markov chain $\mathcal{M}$ if it is a left eigenvector to the eigenvalue 1 of $M$, that is, $qM = q$.

A Markov chain $\mathcal{M} = (V, E, \delta)$ is called irreducible if its graph is strongly connected. The period of a vertex $v$ of a Markov chain is the gcd of all the cycles in $(V, E)$ passing through $v$. A Markov chain is called aperiodic if it has period 1.

The following result is well known and can be found in any standard textbook on the topic.

**Proposition 1.6** We have

i. Every finite Markov chain has a stationary distribution.

ii. Every irreducible and aperiodic Markov chain has a unique stationary distribution.

iii. Every irreducible and aperiodic Markov chain converges to its stationary distribution.

Thus a stationary distribution of an irreducible aperiodic Markov chain $\mathcal{M}$ represents what proportion of time the token will spend in every vertex of $\mathcal{M}$ in the above process.
1.3 Results in the literature

Classical strategic form games were introduced and studied by von Neumann and Morgenstern in their path-breaking work [vNM44], which really established the subject of game theory as a formal branch of mathematics. They proved the famous minmax theorem for two-player strategic form games and showed how to solve such games using linear programming. Nash, in his work [Nas50] proved that every strategic form multiplayer game has a Nash equilibrium in mixed strategies. His proof is of an existential nature and does not provide a procedure to compute the equilibrium. The computation of Nash equilibrium of strategic form games has attracted much attention and work of late. Notable among them are the works of Daskalakis, Papadimitriou and Goldberg [DGP06], Bernhard von Stengel et al [AR SvS10, vS10] etc.

Infinite games on graphs came to the foray mainly after a result of Büchi and Landweber [BL69] in the late sixties who showed that turn-based Muller games played on finite graphs are determined and the winning strategies of either player can be effectively synthesised in finite memory strategies. Martin [Mar75], in a deep result showed that every infinite turn-based game where the winning condition is Borel is determined. Since a Muller condition is also Borel, the determinacy of Muller games follows as a corollary of Martin’s result. Gurevich and Harrington [GH82] cleverly used a data structure called the Latest Appearance Record (LAR) and re-proved the finite-memory determinacy of Muller games. Emerson and Jutla [EJ91] and Mostowski [Mos91] independently showed that turn-based parity games are determined in memoryless strategies. Zielonka [Zie98] used a tree representation of Muller objectives (now known as a Zielonka tree) and presented an elegant analysis of turn-based Muller and parity games. Using an insightful analysis of Zielonka’s result, [DJW97] presented an optimal memory bound for winning strategies in turn-based Muller games. See [Tho90, Tho97] for beautiful surveys on the topic of infinite turn-based games with \(\omega\)-regular winning conditions.

Mean-payoff games have also been widely studied in the literature. [EMT79] showed that such games are memoryless determined. The value of a vertex \(v \in V\), \(val(v)\) of a mean-payoff game is the maximum limit-average reward that the ‘max-player’ can ensure when the play starts at \(v\). A strategy \(s\) of the max-player at a vertex \(v\) is said to be optimal if it attains \(val(v)\). Since such games are memoryless determined, every play finally settles down into a simple cycle and hence it is the cumulative payoffs of the simple cycles that determine the optimal strategies of the players. [ZP96] showed that the
the values of the vertices can be determined and optimal strategies can be effectively synthesised.

For the case of zero-sum multiplayer turn-based games, it has been shown in [CJM04] that every such game with a Borel winning condition has a Nash equilibrium. The existence of subgame perfect equilibrium for multiplayer turn-based games with zero-sum parity objectives was shown in [Umm06, GU08]. In [PS09], we show that in a multiplayer turn-based finite game where the players have preference over the various Muller sets of the vertices, a Nash equilibrium always exists and we give an effective procedure to compute such an equilibrium.

For the case of concurrent games, the literature is not so well developed. The existence of (quantitative) determinacy for concurrent games was shown again by Martin in [Mar98]. Concurrent games with reachability and parity objectives have been studied in [dAHK98, dAH00]. The values of concurrent games with parity objectives was characterised by quantitative $\mu$-calculus formulae in [dAM01]. Chatterjee has extended a few of these results and proved many new ones. See, for instance, his thesis [Cha07] for a nice survey of the existing results on concurrent games and a collection of the results obtained by him.

Logical analysis of strategies in games has been extensively carried out and has a rich literature. Both modal and dynamic logic and temporal logic have been used to reason about strategies.

For the case of finite extensive form games, action indexed modal logics are well suited for logical analysis. Utilities can be coded up in terms of special propositions and the preference ordering is then induced by the implication available in the logic. Game trees themselves are taken as models of the logic. Adopting this approach, a characteristic formula for the backward induction procedure is exhibited in [Bon01]. Progressing further [HvdHMW03] show that the solution concept of subgame perfect equilibrium can be characterised in a modal logic framework. In [vB01, vB02] van Benthem argues that extensive form games can be thought of as process models along with special annotations identifying player nodes. A dynamic logic framework can then be used to describe complete strategies of players as well as reasoning about outcomes that can be ensured. Instead of coding objectives of players in terms of propositions, there have also been suggestions to incorporate elements of modal preference languages into the logic.

Various temporal logics have also been employed to reason about strategies in games. Notable among these is the work on alternating time temporal logic (ATL) [AHK02] which considers selective quantification over paths...
Chapter 1. Games, logic and automata

that are possible outcomes of games in which players and an environment alternate moves. ATL reasons about structured games which are games on graphs where each node is associated with a single normal form game. Since the unfolding of the game structure encodes the past information, the logic itself can be extended with past modalities as well as knowledge modalities in order to reason about the history information and epistemic conditions used in strategising by players (see [vdHW02, JvdH04]). Extensions of ATL where strategies are allowed to be named and referred to in the formulas of the logic are proposed in [vdHJW05] and [WvdHW07]. ATL extended with the ability to specify actions of players in the formulas has been studied in [Ago06] and [Bor07].

Ramanujam and Simon [RS06, RS08b] give a logic to reason about structured strategies which uses the construct $\sigma \sim_i \beta$ which asserts that player $i$ can play according to $\sigma$ and ensure $\beta$. In [RS08a] they consider a dynamic logic in the lines of Parikh’s game logic [Par85] where they reason about game-strategy pairs and their compositions rather than just composing games and analysing strategies separately. Typically, these logical studies involve decidability of satisfiability, complete axiomatisation and (relative) expressiveness of the logics. [Gho08] presents a complete axiomatisation of a logic describing both games and strategies in a dynamic logic framework where assertions are made about atomic strategies.