The Running of the Booles
An Investigation into Boolean Network Games

Declan Thompson
Philosophy
School of Humanities
The University of Auckland

Supervisor: Jeremy Seligman

A dissertation submitted in partial fulfillment of the requirements for the degree of BA(Hons) in Logic and Computation, The University of Auckland, 2015.

Abstract

Boolean Network Games are iterated games played on a network, in which players choose a set of properties to present each round, with the aim of achieving some goal by way of a machine strategy. We define Boolean Games and Iterated Boolean Games as a brief background to the introduction of Boolean Network Games. We then discuss Boolean Network Games in detail, considering their definition and some basic properties. A translation between Boolean Network Games and Iterated Boolean Games is given, grounding the new model in existing frameworks. Finally, we consider a number of extensions to Boolean Network Games to allow for logical representations of machine strategies and the inclusion of epistemology in the games.

Contents

“There are moments, Jeeves, when one asks oneself, ‘Do trousers matter?”’
“The mood will pass, sir.”

The Code of the Woosters,
P.G. WODEHOUSE

Chapter 1

Introduction

Consider the sartorial problem. Every day, we must choose which clothes to wear for that day. There are a finite number of options available, we choose our clothes without knowledge (or, at least, with little knowledge) of what our friends will choose and once we’ve made a selection we’re unlikely to change it for the rest of the day.

Some people may be happy with a random selection of clothes. Some, however, may have some goal in mind. Perhaps they want to fit in with their friends. Maybe they want to stand out from the crowd. They might even want to fit in with their friends, but have their friend group stand out. Each day, they choose their ensemble and then each night they evaluate how well it went. The next day, changes can be made to better reach the goal.

In this dissertation we introduce a model, Boolean Network Games, which aims to capture situations such as this from a game theoretic perspective. In fact the example we have given is a case of general colouring games. Players are arranged in a network (in the example, a friendship network) and have a goal of achieving some property in the network. Such games were studied in an experimental setting in [14] and have applications in modelling social networks, scheduling problems and interaction of automated systems, among others.

In Chapter 2 we introduce Boolean Games and Iterated Boolean Games (IBGs), two models on which Boolean Network Games are based. This provides some background before we introduce Boolean Network Games (BNGs) in Chapter 3. After definition, we discuss some examples of BNGs before exploring general properties in Chapter 4, where we also consider general solutions to a number of Chapter 3’s example games. Chapter 4 is written in a fairly informal style as it aims to present an overview of BNGs without being too caught up in heavy formalisation.

Chapter 5 provides the main technical results of the dissertation, wherein we give two translations which relate BNGs to IBGs. These allow for effective reductions of each type of game to the other and provides a platform for comparing the expressivity of BNGs with that of IBGs. This chapter presents a formal counterpoint to Chapter 4. Chapters 6 and 7 explore a number of modifications to BNGs in order to increase their expressivity. They are intended to probe possible paths of pursuit and hence focus more on informal evaluation of potential definitions and constructions than formal results about them.

Acknowledgements

The author would like to thank Jeremy Seligman for all his help in completing this dissertation. His kind and useful guidance was invaluable in its completion, and I am greatly indebted for his help. I would also like to thank Andrew Withy, Marcus Triplett, Sarah Walker, Mostafa Raziebrahimsaraei, Dennis Dow, Henry Northcott and Sian Thompson for their help with various problems and for proof reading.

Let's start at the very beginning; a very good place to start.

Do-Re-Mi,
RODGERS AND HAMMERSTEIN

Chapter 2

Boolean Games

This chapter introduces a number of basic notions from Boolean Games. Section 2.1 gives a more general overview while Section 2.2 introduces Iterated Boolean Games, which are very closely related to our focus of Boolean Network Games.

2.1 Basic Boolean Games

Boolean Games were first introduced in [11] in a two-player, zero-sum variant. The original construction was demonstrated to be representable in a classical propositional logic, and subsequent discussion of Boolean Games has focussed on this more logical formalisation.

Boolean Games were generalised to n players and (possibly) non-zero sum games in [5], building on work from [7]. In this construction, a Boolean game is a tuple $G = (A, V, \pi, \Phi)$ where $A = \{1, 2 \ldots n\}$ is a set of players, V is a finite set of propositional variables, $\pi : A \to \mathcal{P}(V)$[1] is a *control assignment function* and $\Phi = \{\varphi_1, \varphi_2, \ldots \varphi_n\}$ is a set of Boolean formulas over the variables in V. Intuitively, player i controls the variables in $\pi(i)$, and tries to manipulate them in order to make its goal φ_i true. As such, a strategy for player i is a subset of $\pi(i)$, those variables the player wishes to set to true. Also, we require that π partitions V. That is, $\bigcup_{i \in A} \pi(i) = V$ and if $i \neq j$ then $\pi(i) \cap \pi(j) = \emptyset$.

A strategy profile is a collection of the strategies of each agent, represented by an a n-tuple $S = (s_1, s_2, \ldots s_n)$ where for each i, $s_i \subseteq \pi(i)$. Since a strategy profile gives a valuation on V, we can evaluate formulas with respect to it. Specifically, for $p \in V$ we say $S \models p$ iff $p \in S$ (technically, we should write $p \in s_i$ for some $s_i \in S$, but since the $\pi(i)$s partition V no harm is done). Satisfaction of propositional formulas is given in the standard way. The utility of a strategy profile is given by a Boolean function. For profile S, $u_i(S) = 1$ if $S \models \varphi_i$ and $u_i(S) = 0$ otherwise.

A limitation of Boolean Games is that all players must choose their strategies without any knowledge of the other players' strategies. Hence it is impossible to adjust strategies to other players' moves. The games also only model situations in which players have *perfect knowledge*. That is, all players share the same information about the state of the game.

An extension of Boolean Games which resolves this latter problem is provided in [19]. Here, not only are players' controlled variables restricted, but they are also provided with

[1] $\mathcal{P}(A)$ denotes the powerset of A.

visibility sets, the sets of formulas they can "see". This corresponds with a generalisation of the construction of Boolean Network Games we will present, in which players can only "see" the values of variables in their network neighbourhood. The shortcoming of not being able to adjust to others' strategies is addressed by Iterated Boolean Games, which we introduce in the next section.

2.2 Iterated Boolean Games

Iterated Boolean Games (IBGs) were introduced in [9], where they were used to show how Nash equilibria can be affected by repeated plays of certain games, where each player's strategy can depend upon the choices of other players in the past. Iterated Boolean Games are Boolean Games played infinitely often, with players' strategies able to take into account the past moves of other players. Here we summarise the notion of an iterated Boolean game.

2.2.1 Language

IBGs use the language of Linear Temporal Logic (LTL), $\mathcal{L}_{IBG}$:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi$$

where $p \in \Phi$, a finite set of Boolean variables. [12] gives an overview.

A *run* is a function $\rho : \mathbb{N} \to \mathcal{P}(\Phi)$ that assigns a valuation $\rho[i]$ to every timestep i. $\mathcal{L}_{IBG}$ formulas are interpreted with respect to pairs (ρ, i) where ρ is a run and $i \in \mathbb{N}$. Satisfaction for formulas is defined as follows.

$$\begin{array}{lll}
(\rho, i) \models p & \text{iff} & p \in \rho[i] \\
(\rho, i) \models \neg\varphi & \text{iff} & (\rho, i) \not\models \varphi \\
(\rho, i) \models (\varphi \vee \psi) & \text{iff} & (\rho, i) \models \varphi \text{ or } (\rho, i) \models \psi \\
(\rho, i) \models \mathbf{X}\varphi & \text{iff} & (\rho, i+1) \models \varphi \\
(\rho, i) \models \varphi\mathbf{U}\psi & \text{iff} & (\rho, k) \models \psi \text{ for some } i \leq k \text{ and } (\rho, j) \models \varphi \text{ for all } i \leq j < k.
\end{array}$$

$\mathbf{X}\varphi$ can be interpreted as "φ is true at the ne*x*t timestep" and $\varphi\mathbf{U}\psi$ as "φ is true *u*ntil ψ is true". We say $\rho \models \varphi$ iff $(\rho, 0) \models \varphi$. Intuitively, a run gives a sequence of valuations. The temporal operators are interpreted over this infinite sequence, and we take $i = 0$ to be the "start point".

2.2.2 Games

An *Iterated Boolean Game (IBG)* is a structure

$$G = (A, \Phi, \Phi_1, \dots \Phi_n, \gamma_1, \dots \gamma_n)$$

where $A = \{1, \dots n\}$ is a set of *agents*, Φ is a finite set of Boolean variables, $\Phi_a \subseteq \Phi$ is the set of Boolean variables controlled by agent a, and $\gamma_a \in \mathcal{L}_{IBG}$ is the goal of player a. We require that the sets $\Phi_1, \dots \Phi_n$ partition Φ.

This definition is almost identical to that for Boolean Games. Indeed, the only difference is that the players' goals are formulas of $\mathcal{L}_{IBG}$ rather than basic propositional logic. The major difference with IBGs, however, is in the strategies.

2.2.3 Strategies

Given an IBG $G = (A, \Phi, \Phi_1, \dots \Phi_n, \gamma_1, \dots \gamma_n)$, a *machine strategy* σ_a for player a is an automaton $\sigma_a = (Q_a, q_a^0, \delta_a, \tau_a)$ where Q_a is a finite non-empty set of *nodes*, q_a^0 is the *start node*, $\delta_a : Q_a \times \mathcal{P}(\Phi) \rightarrow Q_a$ is a *transition function* and $\tau_a : Q_a \rightarrow \mathcal{P}(\Phi_a)$ is a *choice* function. Essentially, σ_a gives a valuation at every timestep, dependent on what the other players have chosen at previous timesteps. This allows for strategies to respond to the moves of other players, and increases the range of games possible.

As before, a *strategy profile* is an n-tuple of strategies, one for each player. We denote strategy profiles as $\vec{\sigma} = (\sigma_1, \dots \sigma_n)$, where σ_a is the strategy for player a.

2.2.4 Strategy Induced Runs

A *node vector* of $\vec{\sigma}$ is an n-tuple $\vec{q} = (q_1, \dots q_n)$ where $q_a \in Q_a$ for every $a \in A$. We denote the node vector at timestep i by $\vec{q}[i] = (q_1[i], \dots q_n[i])$. Associated with each node vector $\vec{q}[i]$ is a valuation vector $\vec{v}[i] = (v_1[i], \dots v_n[i])$. These vectors are defined for all timesteps i as follows:

$$\begin{aligned} \vec{q}[0] &= (q_1^0, \dots q_n^0) & \vec{v}[0] &= (\tau_1(q_1^0), \dots \tau_n(q_n^0)) \\ \vec{q}[i+1] &= (\delta_1(q_1[i], \vec{v}[i]), \dots \delta_n(q_n[i], \vec{v}[i])) & \vec{v}[i+1] &= (\tau_1(q_1[i]), \dots \tau_n(q_n[i])). \end{aligned}$$

The *run induced by* $\vec{\sigma}$ is defined as $\rho(\vec{\sigma})[i] = \bigcup_{1 \le a \le n} v_a[i]$, the set of Boolean variables chosen by all the players at each timestep.

2.2.5 Preferences and Nash Equilibria

For each player a we have a preference relation between possible runs given by

$$\rho \succsim_a \rho' \quad \text{iff} \quad \rho' \models \gamma_a \text{ implies } \rho \models \gamma_a$$

If $\vec{\sigma} = (\sigma_1, \dots \sigma_a, \dots \sigma_n)$ and σ_a' is an alternative strategy for a then let $(\vec{\sigma}_{-a}, \sigma_a')$ denote the strategy profile $(\vec{\sigma}_{-a}, \sigma_a') = (\sigma_1, \dots \sigma_a', \dots \sigma_n)$.

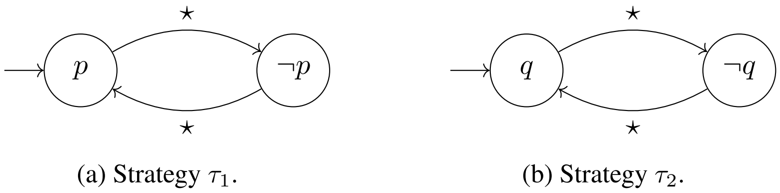
A strategy profile $\vec{\sigma}$ is a *Nash Equilibrium* for a game G if for every player a and every possible strategy $\sigma_a' \in \Sigma_a$ we have $\rho(\vec{\sigma}) \succsim_a \rho(\vec{\sigma}_{-a}, \sigma_a')$. Informally, a cannot do better by changing strategy (assuming all other players' strategies are held constant). In this case, we write $\vec{\sigma} \in NE(G)$.

2.2.6 Example

Consider an example from [10], the IBG defined as

$$G = (\{1,2\}, \{p,q\}, \{p\}, \{q\}, \mathbf{GF}(p \wedge q), \mathbf{GF}(\neg p \wedge \neg q)).$$

G has two players, 1 and 2, and two propositional variables, p (controlled by 1) and q (controlled by 2). The goal of player 1 is that $(p \wedge q)$ should always become true at some future point, i.e., $(p \wedge q)$ should be true infinitely often. Player 2, on the other hand, wants $(\neg p \wedge \neg q)$ to occur infinitely often. It is clear this is not a zero-sum game; both players can achieve their goals at once.



(a) Strategy τ_1.

(b) Strategy τ_2.

Figure 2.1: Two strategies. The transition $\star$ indicates all possible transitions from this node.

Consider the two strategies τ_1, τ_2, depicted in Figure 2.1. In both strategies, the player alternates between setting their variable true and setting it false. Clearly, τ_1 is a strategy for 1 and τ_2 for 2. The run of G with these strategies would look like

$$\{p, q\}, \emptyset, \{p, q\}, \emptyset, \{p, q\}, \emptyset, \ldots.$$

In every even timestep (starting at 0) we have $p \wedge q$, so player 1 achieves its goal. At every odd timestep we have $\neg p \wedge \neg q$, so player 2 achieves their goal. Since neither player can do better, this is a Nash equilibrium.

Emergency! Unknown network detected. Subwave frequency.

Dalek, *The Stolen Earth*,
DOCTOR WHO

Chapter 3

Boolean Network Games

This chapter introduces the notion of a Boolean Network Game, first defined by Seligman and Thompson in [16]. Section 3.1 defines the games and Section 3.2 gives a number of examples.

3.1 Definition of Boolean Network Games

A Boolean Network Game (BNG) is a similar model to an iterated Boolean game but with a network structure on the agents. BNGs are useful for modelling situations in which agents are attempting to find responses to situations with restricted information.

3.1.1 Networks

A *network* $\langle A, R\rangle$ is a finite set A of agents and a binary accessibility relation R on A. For each $a \in A$, the *social neighbourhood* of a is the set $R_a = \{b \mid Rab\}$. We take a finite set of properties PROP. A *local state* is a subset of PROP. A *global state* is a function $g : A \to \mathcal{P}(\mathsf{PROP})$. The *environment of* a, g_a, is the restriction of g to R_a. Intuitively, a local state is the variables a player chooses, and the environment is the variables a player's neighbours have chosen.

Importantly, no restriction is placed on R. We will refer to elements in the social neighbourhood of a as a's neighbours, even though R may not be symmetric.

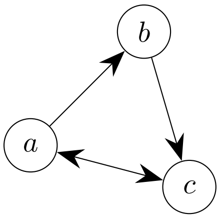


Figure 3.1: A 3-player network. a can "see" both b and c, b can only see c and c only a.

3.1.2 Strategies

A *strategy* for $a \in A$ is a Moore automaton $\langle N, T, I, O\rangle$ where N is a finite set of *nodes*, $I \in N$ is the *start node*, T is a *transition function* mapping nodes and environments of a to nodes and

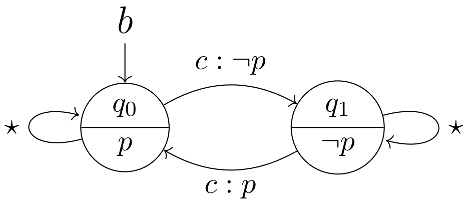


Figure 3.2: The TIT-FOR-TAT strategy.

$O : N \to \mathcal{P}(\mathsf{PROP})$ is an output function mapping nodes to local states of a (subsets of PROP).

Figure 3.2 represents a strategy for player b in the network in Figure 3.1. We call this TIT-FOR-TAT (TiFT). In TiFT, player b starts in state q_0, which has output $\neg p$ (or $\emptyset$). If player c has p, then b transitions from q_0 to q_1 (which outputs p). If b is in q_1 and c has $\neg p$ then b transitions to q_0. The $\star$ transition label indicates that all remaining possibilities use this transition.

A *strategy profile* s is a function mapping each agent a to a strategy $\langle N_{sa}, T_{sa}, I_{sa}, O_{sa} \rangle$. A *node profile* ξ for s is a function mapping each agent a to a node of N_{sa}. The *initial node profile* ξ_{Is} for s is the node profile mapping each agent a to I_{sa}. The *initial global state* g_{Is} is the global state mapping each agent a to $O_{sa}(I_{sa})$.

Suppose s is a strategy profile and ♲ is a strategy for a. The *modification* of s with ♲ for a is the function

$$s_{a:♲}(b) = \begin{cases} ♲ & b = a \\ s(b) & b \neq a. \end{cases}$$

3.1.2.1 General Strategies

A strategy depends inherently on the neighbourhood of the player it is for. Thus a strategy needs to be defined for a particular agent, given its neighbourhood. In some cases we may wish to provide more generalised strategies, rather than specifying a different strategy for each neighbourhood. We define *general strategies*, which are classes of strategies, containing a strategy for each possible neighbourhood of a player.

Figure 3.3 demonstrates the specification of a general strategy. Generalised TAT-FOR-TIT (TaFT$_G$) is the strategy where each player changes state iff all of its (non-empty) neighbours are the same colour as it. A player's initial node is indicated by the relevant start state.

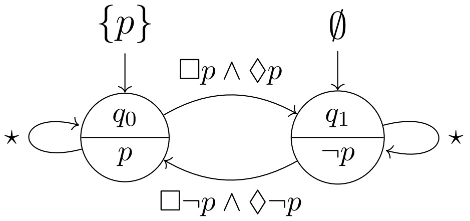

$\{p\}$
$\emptyset$
$\Box p \wedge \Diamond p$
q_0
p
q_1
$\neg p$
$\Box \neg p \wedge \Diamond \neg p$
⋆
⋆

Figure 3.3: The Generalised TAT-FOR-TIT strategy.

3.1.3 Outcomes

Given a global state g and a state profile ξ for s, the *next node profile* $\xi_{s,g}$ and the *next global state* are given by

$$\xi_{s,g}(a) = T_{sa}(\xi(a), g_a) \qquad g_{s,\xi}(a) = O_{sa}(T_{sa}(\xi(a), g_a))$$

These are the profiles after a single round of interaction between the agents. The *outcome behaviour* of s is the infinite sequence $\{\langle g^i, \xi^i\rangle\}_{i\in\mathbb{N}}$ defined by

$$\langle g^0, \xi^0\rangle = \langle g_{Is}, \xi_{Is}\rangle \qquad \langle g^{i+1}, \xi^{i+1}\rangle = \langle g^i_{s,\xi^i}, \xi^i_{s,g^i}\rangle$$

The sequence $g^0, g^1, g^2 \ldots$ describes the evolution of the agents' properties over time. The sequence $\xi^0, \xi^1, \xi^2, \ldots$ describes the evolution of the agents' internal nodes over time.

3.1.4 Language

We use an extension of Linear Temporal Logic (LTL) called $\mathcal{L}_{BNG}$ to allow us to describe the network relation over time.

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \Box\varphi \mid X\varphi \mid \varphi U \varphi$$

where $p \in \mathsf{PROP}$. These propositions express properties at each agent. For example, $\Box p$ says that all my neighbours have property p.

A *network model* $M = \langle A, R, g\rangle$ is a network $\langle A, R\rangle$ with a global state g. Formulas are evaluated with respect to a strategy profile s for the network, an agent $a \in A$ and a timestep i as follows:

$$\begin{array}{lll}
M, s, a, i \vDash p & \text{iff} & p \in g^i(a) \\
M, s, a, i \vDash \neg\varphi & \text{iff} & M, s, a, i \nvDash \varphi \\
M, s, a, i \vDash (\varphi \vee \psi) & \text{iff} & M, s, a, i \vDash \varphi \text{ or } M, s, a, i \vDash \psi \\
M, s, a, i \vDash \Box\varphi & \text{iff} & M, s, b, i \vDash \varphi \text{ for all } b \in R_a \\
M, s, a, i \vDash X\varphi & \text{iff} & M, s, a, i+1 \vDash \varphi \\
M, s, a, i \vDash \varphi U \psi & \text{iff} & M, s, a, k \vDash \psi \text{ for some } i \leq k \\
 & & \text{and } M, s, a, j \vDash \varphi \text{ for all } i \leq j < k.
\end{array}$$

We say that $M, s, a \vDash \varphi$ iff $M, s, a, 0 \vDash \varphi$.

3.1.5 Games

Given a network model $M = \langle A, R, g\rangle$, a *goal profile* is a function $\gamma : A \to \mathcal{L}_{BNG}$. A Boolean network game (BNG) is a pair $G = \langle M, \gamma\rangle$. For any player $a \in A$, a strategy for a $\langle N, T, I, O\rangle$ is *available to* a iff $O(I) = g(a)$.

The *utility* of a strategy profile s for a is given by

$$u_a(s) = \begin{cases} 1 & \text{if } M, s, a \vDash \gamma(a) \\ 0 & \text{otherwise} \end{cases}$$

A strategy profile s is a *Nash Equilibrium* if there is no player a and strategy ♲ for a such that $u_a(s_{a:\text{♲}}) > u_a(s)$. That is, if no player can do better by choosing a different strategy (while all other players' strategies are kept constant). In this case, we write $s \in NE(G)$.

3.2 Examples

An obvious class of games to consider on networks is that of colouring games. In these games, local states represent colours and the goal of each player relates to ensuring the same or different colouring to its neighbours. The colouring games described in [14] are good examples.

We can restrict colouring games to those in which every player has the same goal. Under this assumption, there are two broad classes of games we can consider - conformity games, where all players try to be the same colour, and diversity games, where players try to be a different colour to their neighbours. For simplicity, we will consider games where $\mathsf{PROP} = \{p\}$. For colouring, this means there are two colours available. Table 3.1 gives some colouring goals players may have.

$$\begin{aligned}
\mathsf{Con} &= ((p \wedge \Box p) \vee (\neg p \wedge \Box \neg p)) & \mathsf{Div} &= ((p \wedge \Box \neg p) \vee (\neg p \wedge \Box p)) \\
\mathsf{C1} &= (p \leftrightarrow \Box p) & \mathsf{D1} &= (p \leftrightarrow \Box \neg p) \\
\mathsf{C2} &= (p \leftrightarrow \Diamond p) & \mathsf{D2} &= (p \leftrightarrow \Diamond \neg p) \\
\mathsf{NCon} &= (\Box(p \wedge \Box p) \vee \Box(\neg p \wedge \Box \neg p)) & \mathsf{NDiv} &= (\Box(p \wedge \Box \neg p) \vee \Box(\neg p \wedge \Box p))
\end{aligned}$$

Table 3.1: Some colouring goals.

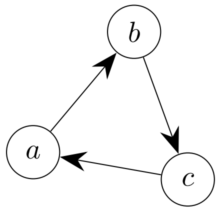
In a conformity game, players aim to be the same colour as their neighbours. There are a number of potential formulas which specify the goal for this game. A player achieving Con requires both that all their neighbours are monochromatic, and that they are the same colour as their neighbours. Weaker versions of Con include $\mathsf{C1}$, in which players have $\neg p$ unless all their neighbours have p, and $\mathsf{C2}$, in which players have p unless all their neighbours have $\neg p$. NCon is the goal that a player's neighbours all have Con.

In a diversity game, players aim to be a different colour to their neighbours. Again, there are many ways to express this goal. Div is the analogue of Con - a player has Div if all their neighbours are monochromatic, and the player is a different colour to them. $\mathsf{D1}$ requires a player has $\neg p$ unless all its neighbours have p, and $\mathsf{D2}$ requires a player has p unless all its neighbours have $\neg p$. Like NCon, NDiv requires all neighbours to have Div.

Our goals do not include temporality. To incorporate this, we can add temporal modalities. We will use the standard defined modalities $F\varphi = (\top U \varphi)$ and its dual $G\varphi = \neg F \neg \varphi$. Intuitively, F means "at some *f*uture point" and G is read as "it's always *g*oing to be that".

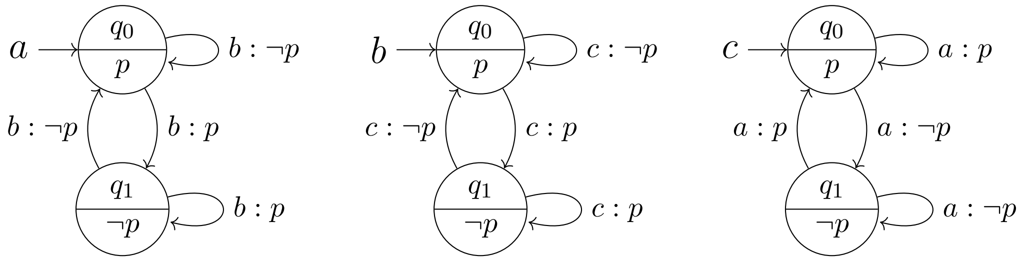
3.2.1 3-Cycle Unstable Colouring

Consider the game G_{3CU} described in Figure 3.4. The goals are our standard diversity goals - each player wishes to be a different colour from those players it can see. Player a wishes to be different to b, who wishes to be different to c, who in turn wishes to be different to a. In this example, each player's goal is equivalent to $GF\mathsf{D1}$ and $GF\mathsf{D2}$. We call this goal profile the *unstable colouring* goal profile. This is because each player's goal is to achieve colouring infinitely many times (GF), but not necessarily permanently. Every player's start state is $\{p\}$, and we will assume that $\mathsf{PROP} = \{p\}$.



$$\begin{array}{ll}\gamma(a) = GF\mathsf{Div} & g(a) = \{p\} \\ \gamma(b) = GF\mathsf{Div} & g(b) = \{p\} \\ \gamma(c) = GF\mathsf{Div} & g(c) = \{p\}\end{array}$$

Figure 3.4: G_{3CU}. 3-cycle unstable colouring game.



(a) TAT-FOR-TIT (b) TAT-FOR-TIT (c) TIT-FOR-TAT

Figure 3.5: Strategy profile s. Players a and b use TaFT and c uses TiFT.

Figure 3.5 presents a strategy profile s that we can use in G_{3CU}. In s, both a and b change to the opposite of the player they can see each round. The idea is that the observed player might have their state fixed, and so the goal can be met in the next round. Call this strategy TAT-FOR-TIT (TaFT). In contrast, c chooses the same colour as its neighbour at each round, the TIT-FOR-TAT (TiFT) strategy.

Now consider the outcome behaviour of s in G_{3CU}. Table 3.2a gives the evolution of the global state, and Table 3.2b gives the evolution of the strategy node profile. Though infinite, the run loops. To give a more intuitive understanding of the run, we can represent the global state evolution graphically, as in Figure 3.6. This notation works only for games with $|\mathsf{PROP}| = 1$.

We may ask if any players achieve their goal under s. We have enough of the outcome behaviour to determine the answer - since the strategy node profiles have looped, we have seen everything that is possible under s. If a player achieves its goal then it must do so within the loop, which in this case means within $0, 1, 2$. Each player wants to have a different value of p to their neighbour in at least one of these timesteps. a achieves this at timestep 2, b at timestep

	0	1	2	3	4	5	...
a	p	$\emptyset$	p	p	$\emptyset$	p	...
b	p	$\emptyset$	$\emptyset$	p	$\emptyset$	$\emptyset$	...
c	p	p	$\emptyset$	p	p	$\emptyset$	...

(a) Evolution of global states

	0	1	2	3	4	5	...
a	q_0	q_1	q_0	q_0	q_1	q_0	...
b	q_0	q_1	q_1	q_0	q_1	q_1	...
c	q_0	q_0	q_1	q_0	q_0	q_1	...

(b) Evolution of strategy nodes

Table 3.2: Outcome behaviour of s.

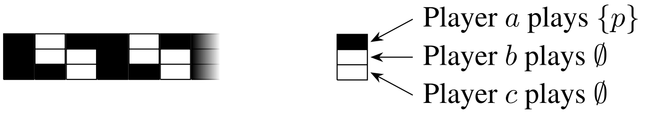
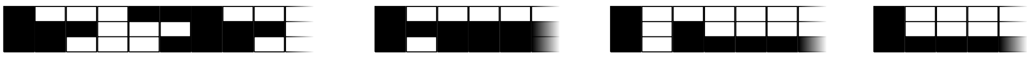


Figure 3.6: Graphical representation of the global state evolution of s.



(a) $s_{b:\text{TiFT}}$ in G_{3CU} (b) t in G_{3C} (c) $t_{b:\text{TaFT}}$ in G_{3C} (d) u in 3_{B3C}

Figure 3.7: Outcome behaviours of various strategies in games.

1 and c at both timesteps 1 and 2. In summary,

$$u_a(s) = 1 \qquad u_b(s) = 1 \qquad u_c(s) = 1.$$

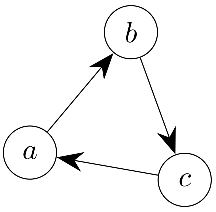
Clearly this is a Nash Equilibrium, since every player achieves its goal. Now what if b uses TiFT as well as c?

Figure 3.7a gives the outcome behaviour of $s_{b:\text{TiFT}}$ in G_{3CU}. (We have not given the node profiles, since each node gives a unique output in TiFT and TaFT.) Which players achieve their goal this time? For each player, we can check by determining if that player differs from the player below it at any timestep. Player a achieves its goal at timesteps $1, 2, 4, 5$. Player b achieves its goal at timesteps 2 and 5. And player c achieves its goal at timesteps 1 and 4. Every player achieves its goal, and the run loops, so this is also a Nash equilibrium.

3.2.2 3-Cycle Stable Colouring

Let us modify G_{3CU} to G_{3C} by changing the goal profile, as indicated in Figure 3.8. These goals differ from 3_{GCU} in the ordering of F and G. In G_{3C} players are concerned with reaching a stable state of diversity, hence this is the *stable colouring* goal profile. Clearly there is no way for every player to reach its goal, otherwise we would be able to 2-colour the triangle.

The network has not changed, so the strategy profiles σ and $\sigma_{b:\text{TiFT}}$ still work, and will have the same run as in G_{3CU}. Since both profiles have a repeat of ■ in their runs, no player achieves their goal under either. Instead, consider a new strategy profile t, described in Figure 3.9. In t, c uses the TaFT strategy. a plays p unless b also plays p, in which case a plays $\neg p$ forever after, the TWEETYPIE strategy. b stubbornly remains as p (and so a will change to $\neg p$).



$a:$	$F G\mathsf{Div}$	$g(a) = \{p\}$
$b:$	$F G\mathsf{Div}$	$g(b) = \{p\}$
$c:$	$F G\mathsf{Div}$	$g(c) = \{p\}$

Figure 3.8: G_{3C}. 3-cycle stable colouring game.

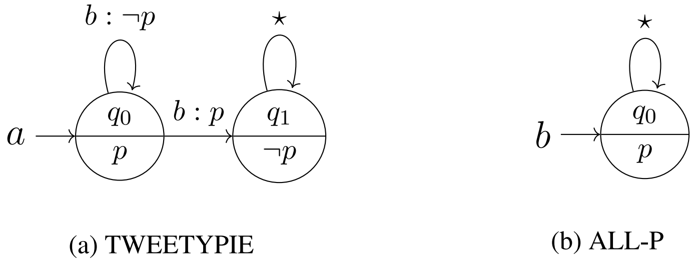


Figure 3.9: Strategies for t. Player c uses TaFT.

Figure 3.7b shows the outcome of t in G_{3C}. We end up in a stable state, so to determine which player achieves its goal we need only evaluate at timestep 2. Here a and c achieve their goals, but b does not, since it eventually stays the same "colour" as c. Thus the only player who can do better is b. Indeed if b adopts TaFT, it can do better, as shown in Figure 3.7c. Again we reach a stable final state, but now b and c achieve their goals, and a does not. So t is not a Nash equilibrium, since b could do better. A natural question is whether G_{3C} has a Nash equilibrium. In section 4.1.2 we will show that it does not.

3.2.3 Broken 3-Cycle Stable Colouring

Now consider a modification of the 3-cycle network by reversing the direction of one of the arrows, as shown in Figure 3.10a. For G_{B3C} we will keep the same stable colouring goal profile, and the same initial state profile (every player has $\{p\}$).

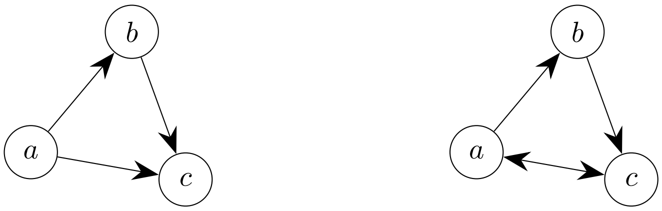
Now a has two neighbours, so none of the strategies we have been using are available to it, since their transitions require exactly one neighbour. Instead, let us assign a TaFT$_G$ (described in Figure 3.3). Player b is able to utilise 1-neighbour strategies, so assign TaFT to b. Player c has no neighbours. $FG\mathsf{Div}$ contains the $\Box$ modality, which becomes trivially true whenever a player has no neighbours. Because of this, c will achieve its goal no matter what strategy it plays! Let us assign ALL-P to c (a generalised version of ALL-P is described in Figure A.1).

Call this strategy profile u. Figure 3.7d gives the outcome behaviour of u in G_{B3C}. The run stabilises quickly, with c the only player playing p. Clearly b and c succeed in their goals (c trivially so) but a does not, since though it is coloured differently to c it is still the same colour as b. In fact a *cannot* do better. Neither b nor c is affected by the moves a makes (more formally, a is not part of the generated subgraph rooted at either b or c), and so no matter a's strategy b will always (eventually) be coloured differently to c, given its strategy. So u constitutes a Nash equilibrium.

3.2.4 Modified 3-Cycle Stable Colouring

Let's modify the network again to give the graph in Figure 3.10b. Now the edge between a and c is bidirectional. Also, we'll take the following new goal profile.

$$\begin{aligned} \gamma(a) &= FG(\Box p \lor \Box\neg p) & \qquad g(a) &= \{p\} \\ \gamma(b) &= FG\mathsf{NDiv} & \qquad g(b) &= \{p\} \\ \gamma(c) &= FG\mathsf{Con} & \qquad g(c) &= \{p\} \end{aligned}$$



(a) Broken 3-cycle for G_{B3C} (b) Modified 3-cycle for G_{M3C}

Figure 3.10: Further 3 player networks.

Player a wants b and c to be the same colour, b wants a and c to be different and c wants to be the same as to a. Call this game G_{M3C}. We will adopt a new strategy profile v for this game. Under v, a uses the WATCH-AND-WAIT strategy, shown in Figure 3.11. We assign $v(b)$ = TaFT (the opposing strategy) and $v(c)$ = TiFT (the copying strategy). Figure 3.12a gives the run of v in G_{M3C}. As can be seen, from timestep 3 onwards b and c are coloured identically, so a achieves its goal. Also from 3, a and c are coloured differently, so b achieves its goal. Player c does not.

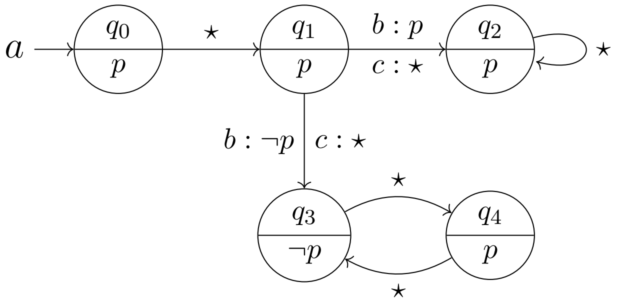
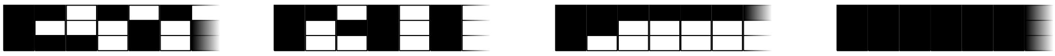


Figure 3.11: The WATCH-AND-WAIT strategy.

v is not a Nash equilibrium. Indeed, c can do better by choosing TaFT at b's expense, as seen in Figure 3.12b. From this, b can do better by choosing TiFT (Figure 3.12c). But then c can do better by swapping back to TiFT (Figure 3.12d). And we reach a loop back to v, since b can do better by reverting to TaFT.

Notice that in every run in Figure 3.12, a achieves its goal. Is WATCH-AND-WAIT then a winning strategy for a in general? Certainly if b and c's choices are restricted to TiFT and TaFT it is. Unfortunately there are trivial counter-examples to WATCH-AND-WAIT being always winning. If b adopts ALL-P and c adopts NEVER-P, a will not win (though b and c probably won't either). In the next chapter we will examine general solutions in more detail.



(a) v in G_{M3C} (b) $v_{c:\text{TaFT}}$ in G_{M3C} (c) $v_{b:\text{TiFT},c:\text{TaFT}}$ in G_{M3C} (d) $v_{b:\text{TiFT}}$ in G_{M3C}

Figure 3.12: Various modifications of v on G_{M3C}.

> Democracy is when the indigent, and not the men of property, are the rulers.
>
> ARISTOTLE

Chapter 4

Properties of Boolean Network Games

In this chapter we discuss in more detail certain aspects of Boolean Network Games. We will take an informal style in order to present an overview of interesting features. In Section 4.1 we discuss some preliminary results about BNGs in general. Many of these results will be useful in later chapters, particularly Chapter 5, in which we establish an equivalence between BNGs and Iterated Boolean Games. Section 4.2 deals with the task of finding general solutions to certain goals. Is there a certain strategy which guarantees a certain goal is achieved? Can network conditions help?

4.1 General Results

This section discuss two interesting general results about BNGs. The first deals with the notion of *myopic strategies*, which will be very useful in later chapters. The second identifies a benefit of having only one neighbour in a game.

4.1.1 Double Vision and Myopia

The reader will have noticed that in all the examples in Chapter 3 the games' runs eventually looped. In some cases, the loop started immediately, while in others there was some “fluff” before the loop began. This looping behaviour was not accidental, and occurs in every BNG run. The following lemma describes this fact, and is a crucial aspect in the proofs of a number of subsequent results.

Lemma 1 (Looping Lemma)**.** *The run of every BNG on any strategy eventually repeats.*

Proof. First notice that since strategies are deterministic and each strategy node profile determines a global state, given a node profile the next node profile is uniquely determined (that is, the outcome behaviour is deterministic). Since BNGs contain finitely many players, and each player's strategy contains finitely many nodes, there are finitely many possibly strategy node profiles, given a BNG and a strategy profile. So in the infinite run, some node profile must be repeated, and as soon as it is we are in a loop, since subsequent profiles are deterministic. ⏢

The looping lemma is very similar to a famous result in the theory of Büchi automata [1]. A Büchi automaton is like a deterministic finite automaton (see [17] for an overview), but which runs on infinite strings. Büchi's Theorem states that every Büchi-recognisable language is the finite union of a set of strings of the form $w \cdot v^\omega$. This is very similar to what the looping lemma says; the initial prefix w is the "fluff", and the string v is the looping component.

Another concept which will be very useful is that of a *myopic* strategy. Intuitively, a strategy is myopic if the behaviour of its neighbours has no impact on its evolution. In other words, there is exactly one transition from each node. This type of strategy is described for Iterated Boolean Games in [10], and can be more formally stated by the following definition.

Definition 1 (Myopic Strategy)**.** *A BNG strategy $\langle N, T, I, O\rangle$ is myopic iff $T(v, g_a) = T(v, g'_a)$ for every node v and environment g_a, g'_a.*

A myopic strategy can be seen as a run of valuations that eventually loops. When specifying myopic strategies, we will take the transition function T to have one argument, since the environment will never have an impact. To see the usefulness of myopic strategies, let us introduce a notion of strategic equivalence.

Definition 2 (Game Equivalent)**.** *Let G be a BNG, s a strategy profile for G, ♲, ♲$'$ strategies for player $a \in A$. Strategy ♲ is* game-equivalent *to ♲$'$ if for every player $b \in A$ and timestep i we have*

$$M, s_{a:♲}, b, i \models \varphi \quad \textit{iff} \quad M, s_{a:♲'}, b, i \models \varphi.$$

Essentially, ♲ and ♲$'$ are game equivalent if they behave the same way in a particular game and strategy profile. The following lemma connects the notions of myopic strategies and game equivalence, and will be very important for the equivalence results established in Chapter 5.

Lemma 2 (Existence of equivalent myopic strategies)**.** *Let G be a BNG, s a strategy profile for G and $a \in A$ a player. Then there is a myopic strategy which is game equivalent to $s(a)$ under G and s.*

Proof. By the Looping Lemma, the run of s on G eventually loops. That is, the outcomes of s on G look like $g^0, g^1 \ldots g^i, g^{i+1}, \ldots g^k, g^i, g^{i+1}, \ldots$. Define a myopic strategy ♲ for a by setting $N_{♲} = \{j \in \mathbb{N} \mid 0 \leq j \leq k\}$, $T_{♲}(j) = j + 1$ for $0 \leq j < k$, $T_{♲}(k) = i$, $I_{♲} = 0$ and $O_{♲}(j) = g^j(a)$.

The strategy ♲ is myopic and outputs exactly the same sequence of states as $s(a)$ does in the run of s on G. Every other strategy in s will be fooled, and so ♲ is game equivalent to $s(a)$ under g and s. ∎

Although strategies can be replaced by myopic modifications, important properties like Nash equilibria are not preserved. If a changes to a myopic strategy then other players may be able to take advantage of a's predictability and inability to retaliate. The equivalence provided by game equivalence thereby is quite weak; game equivalence requires that nothing else in the game changes, that every other player keeps its strategy the same. A strengthened notion is given by *strong* equivalence.

Definition 3 (Strongly Equivalent)**.** *Two strategies ♲, ♲$'$ for a are* strongly equivalent *if for every BNG containing a and strategy profile, ♲ is game-equivalent to ♲$'$.*

Game equivalence allows us to swap strategies when all other aspects of a game are fixed. Contrastively, strong equivalence tells us that the strategies behave the same way under any strategy profile in any game. It follows from this that Nash Equilibria are preserved under strong equivalence.

Proposition 1. *Let G be a BNG, s be a strategy profile for g and $♲, ♲'$ be two strongly equivalent strategies for a. Then $s_{a:♲} \in NE(G)$ iff $s_{a:♲'} \in NE(G)$.*

Proof. If $s_{a:♲} \notin NE(G)$ then some player b can do better. If $b \neq a$ then ♲ and $♲'$ behave the same way no matter what strategy b uses, so $s_{a:♲'} \notin NE(G)$. If $b = a$ then since ♲ and $♲'$ give the same outcomes, a can also do better than $♲'$. ∎

4.1.2 Strategies for Degree 1 Players

Recall the three player stable colouring game G_{3C} from section 3.2.2, as described in Figure 4.1. We asked if there a Nash equilibrium for this game. In order to answer this question, consider a smaller game containing two players x and y, both of whom can see the other. Suppose that, as G_{3C}, both players have the goal $FG\mathsf{Div}$. Then we make the following claim.

Proposition 2. *Player x can always find a winning strategy, whatever player y's strategy is.*

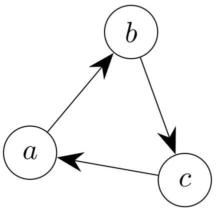
Proof. Construct a myopic strategy for x as follows. Begin at node q_0, with the required output. Determine y's next node q_y at the next timestep, given its current strategy. If q_y has already not already been visited, create a new node and set its output to the opposite of q_y's. Otherwise, loop back to the node corresponding to q_y.

At every timestep (except possibly the first), x has a different output to y, so x has a winning strategy. ∎

How does this help with our question of Nash equilibria in the 3-cycle problem? We know that in any strategy profile at least one player has a losing strategy (otherwise we could 2-colour the triangle). Call this player x, and the other two players y_1 and y_2, where x can see y_1 and y_2 can see x. From x's perspective, y_1 and y_2 can be treated as a single player, with a strategy as the composition of y_1 and y_2's strategies. By Proposition 2, x has a winning strategy. Hence no strategy profile can be a Nash equilibrium. This result can be generalised, as in Proposition 3.

Proposition 3. *If x is a player in some BNG such that $|R_x| = 1$ then for any strategy profile, x has a winning strategy for the goal $FG\mathsf{Div}$.*

Proof. Given the strategy profile, construct a myopic strategy for x by using the same method from Proposition 2. ∎



$a:$	$FG\mathsf{Div}$	$g(a) = \{p\}$
$b:$	$FG\mathsf{Div}$	$g(b) = \{p\}$
$c:$	$FG\mathsf{Div}$	$g(c) = \{p\}$

Figure 4.1: G_{3C}. 3-cycle stable colouring game.

4.2 General Solutions

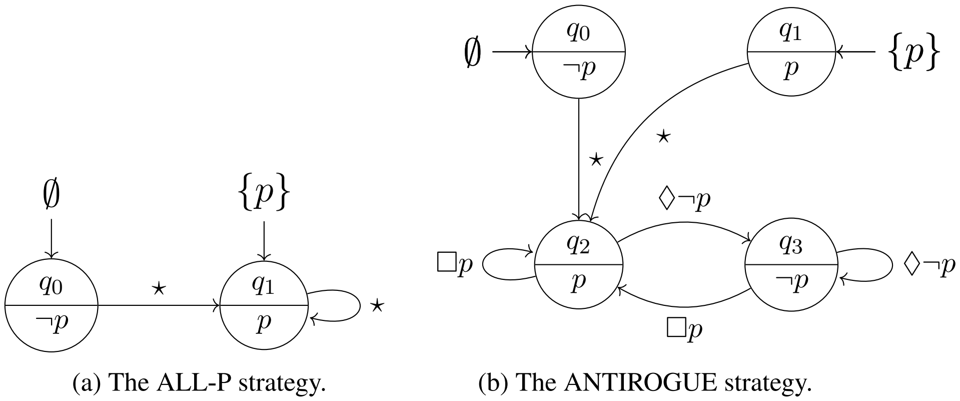
Given a certain goal, a natural question is to ask whether there is a strategy which will guarantee a player achieves that goal in all games, whatever every other player's strategy is. Certainly this is the case for some goals. For example, if a player's goal is a tautology, then any strategy will be successful. Slightly less trivially (but still quite uninteresting), if a goal is of the form $FG\varphi$, where φ is a non-contradictory propositional formula (no use of the modalities $\Box$, X or U), we can easily construct a strategy to achieve it by changing to a satisfying valuation as soon as possible.

Of course, achieving these goals is not unexpected. They make limited use of the temporal nature of games, and no use of the network structure. Even addressing the temporal aspect is fairly straightforward, by incorporating standard results from linear temporal logic. Kamp's Theorem states that LTL is equivalent to the monadic first-order language of order [15, 13]. This was shown in [18] to be equivalent to ω-regular languages without the $\star$ operator. Büchi showed that Büchi automata are able to recognise ω-regular languages [2]. In fact, for any satisfiable LTL formula (note, no network modalities) there is a strategy profile which will satisfy this formula. This result follows immediately from a similar result in [10] about Iterated Boolean Games.

Instead, we focus on the network structure. Here it becomes more difficult to guarantee achieving a goal, since this may depend on the behaviour of other players. In order to consider how network structures can impact the solution of *general solutions*, that is, strategies which guarantee a player achieves its goal, we will consider again the colouring examples from Chapter 3.

4.2.1 General Solutions to the Conformity Game

Games like stable conformity can be easily solved by trivial general strategies. For example, we can have every player choose p as soon as they can (ALL-P, see Figure 4.2a). This solution is somewhat disappointing and unenlightening. We should like to refine the problem in some way, in order to rule this solution out.



(a) The ALL-P strategy. (b) The ANTIROGUE strategy.

Figure 4.2: General solutions for conformity.

Dubey and Shapely [6] faced a similar problem in their analysis of trading economies: "For example consider the case where each individual bids and supplies nothing". In order to consider "non-pathological" equilibria they take a situation in which an outside agency places a fixed bid of $\epsilon > 0$. That is, some agent disrupts the trivial equilibrium by bidding and supplying something. This allows for "nice" equilibria, which are the limit equilibria as $\epsilon \to 0$ (and the player removes itself). We could take a similar approach by supposing that some player will always choose a random state, and stay fixed in it. This would solve the problem for the conformity game - we are looking for a general strategy that work assuming some unknown player fixes a random colour. Clearly, any solution will involve changing to this player's colour. We probably need to add the restriction that the unknown player must be reachable in the network from every player, so as to exert some influence.

Unfortunately this approach results in similar trivial solutions. The ANTIROGUE general strategy in Figure 4.2b allows every player to win, even when a rogue (but still accessible) player is introduced. In this strategy, whatever a player's initial start state is, they immediately change to p. If they then have a neighbour which is $\neg p$, that neighbour must be the rogue player, and so they change to that player's colour.

Again, this solution is somewhat trivial. Potentially, we allow the rogue player to have an unknown strategy of bounded size. Then the game becomes "given an unknown rogue with strategy of size at most n, what general strategy will achieve conformity?". This approach still seems rather ad hoc. And in fact, it seems plausible that similar strategies to ANTIROGUE will be able to circumvent bounded size strategies - there are only finitely many such strategies to consider.

Another way to avoid these trivial solutions may be to rule out the use of *predictable* strategies.

Definition 4 (Predictable Strategy). *A strategy is* predictable *if its outcome behaviour does not depend on the game's network structure or the strategies of the other players.*

All myopic strategies are predictable. We can construct strategies which are predictable but not myopic by splitting myopic strategies to depend on the output of other strategies. However, these split strategies are still in some way equivalent to the myopic strategies. Specifically, we have the following proposition.

Proposition 4. *Every predictable strategy is strongly equivalent to a myopic strategy.*

Proof. If ♲ is predictable then we know its outcome behaviour in any game. Since ♲ is finite, we can use this outcome behaviour to build a myopic strategy strongly equivalent to it. ∎

In order to rule out these trivial cases, we can pose a more interesting question. Is there an unpredictable general strategy which leads to conformity on any network? This question still doesn't quite get there. Notice that ANTIROGUE is an unpredictable strategy, since its behaviour depends on the existence and behaviour of a rogue. Perhaps a better refinement would be to require the outcome to depend on the initial state of the players - we want to conform to whatever was the majority initially.

Finding a good refinement of the conformity problem which rules out trivial solutions is an open problem.

4.2.2 General Solutions to the Diversity Game

We can ask similar questions to conformity of diversity. Is there a general strategy that will ensure stable diversity in a given network? Here there are no obvious trivial solutions to exclude. However, we must note that, unlike conformity, diversity cannot always be achieved. Specifically, it is productive to restrict our question to networks based on bipartite graphs. However, even with this restriction we obtain a negative result to our primary question.

Proposition 5. *There is no general strategy which ensures diversity on arbitrary bipartite networks.*

Proof. Take the cycle graph C_{2n} for $n \geq 2$. Suppose every player is initially coloured white. Since every player has the same general strategy, and every player can see exactly two other players, every player has the same strategy. Hence, every player will change colours in synchronisation with every other, so diversity will never be achieved. ⏢

This leads us to a new question. Which networks *do* have a general strategy for diversity? We can understand this question in two ways. First, what is the class of networks for which general strategies for diversity exist (where the general strategy for each class may be different)? Second, given a general strategy, what is the class of networks upon which this guarantees diversity?

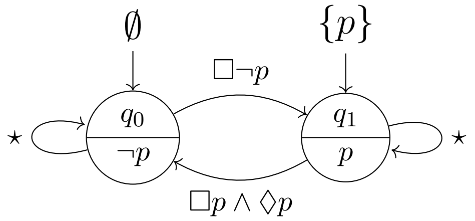


Figure 4.3: The p-Stable TAT-FOR-TIT strategy.

Certainly for some classes of graphs we are able to find general strategies for diversity. Take, for example, the class of perfect trees (that is, rooted trees where every leaf has the same distance from the root). For this class of graphs, the p-stable TAT-FOR-TIT general strategy (see Figure 4.3) gives a solution. This is the strategy of changing to p if all your (possibly empty) neighbours have $\neg p$, changing to $\neg p$ if all your (non-empty) neighbours have p and staying the same otherwise. Since leaves have no neighbours, they will always see $\Box\neg p$, so all leaves will be p (and stay p) after the first round. Then the neighbours of leaves will all see $\Box p \wedge \Diamond p$, so will change to $\neg p$, and so on. Eventually, the root will change to p if the tree is of even height, and $\neg p$ if odd.

4.2.3 General Solutions to Unstable Games

In the previous sections, we considered the stable diversity (colouring) and conformity games. But what about unstable colouring games? If players have the goal $GF\mathsf{Div}$, is there a general strategy which achieves this? For similar reasons as before (the C_{2n} counterexample for one), the answer is no. We are, however, able to do better with unstable colouring than with stable colouring. In fact it is possible to achieve unstable colouring on any graph, whether bipartite or not.

Proposition 6. *Every graph has a Nash equilibrium for unstable colouring.*

Proof. Define a linear order on the powerset of the players $\mathcal{P}(A)$ (which is finite in size). For each player, define a myopic strategy in which each node corresponds to an element of $\mathcal{P}(A)$, the nodes are visited as per the linear order, and the output of a node is p iff the node contains the current player.

Every possible configuration is visited at least once. So every player achieves its goal at least once in every cycle. Since every player achieves its goal, this is a Nash equilibrium. ∎

In fact, this result can be generalised to any network, and to any goal of the form $FG\varphi$, where φ contains no temporal operators.

Proposition 7. *If every player a in a BNG G has a goal of the form $GF\varphi$, where φ is a modal formula satisfiable at a, then there is a Nash equilibrium for G in which every player has a winning strategy.*

Proof. This follows by a similar argument to Proposition 6. ∎

Between the idea and the word
there is more than we can understand.
There are ideas for which no words can be found.

The thought lost in the eyes of a unicorn
appears again in a dog's laugh.

Between, VLADIMÍR HOLAN

Chapter 5

Expressivity of Boolean Network Games

Boolean Network Games and Iterated Boolean Games are similar structures with a differing basis. Where IBGs add a temporal structure to what is essentially a propositional base, BNGs add this temporality to a *modal* base.

Standard results allow us to model basic modal logics inside predicate logic (see, for example, the discussion on the Standard Translation in [4, pp.83-90]). Propositional logic is ill-suited to this task however, with its lack of a relational structure. Even so, the similarity of BNGs to IBGs presents a natural question: Can BNGs be modelled by IBGs? That is, can we translate any BNG into an IBG in a way which preserves the impact the accessibility relation has on the interaction of the agents? In practice, the accessibility relation imposes a restriction on the *transition functions* of players' strategies. While a propositional setting cannot encode modal relations on its own, perhaps a restriction of transition functions can achieve the same ends.

We can also ask the converse question. Given an IBG, can we model it as a BNG? At first this seems an easy prospect. Take a complete graph for the relation, ensuring every player can see every other, and proceed as normal. But we quickly encounter problems. In a BNG every player has control over all the propositional variables; in an IBG, each player has control over only a subset, and different players may control different numbers of propositional variables. Perhaps a player can achieve more by changing other players' variables?

In this section we propose two translations, first from BNGs to IBGs and second from IBGs to BNGs. For each translation, we consider what properties of games are preserved. Finally, we give some results on the complexity of certain decision problems related to BNGs.

5.1 Translation from Boolean Network Games to Iterated Boolean Games

In this section, we give a translation from Boolean Network Games to Iterated Boolean Games. By abuse of notation, we use $\mathcal{T}$ for all functions related to the translation; which function is intended will be clear from context.

5.1.1 Game Translation

Suppose we have a BNG $G = \langle M, \gamma \rangle$, where $M = \langle A, R, g \rangle$ and $A = \{1, 2, \dots n\}$, and that $\mathsf{PROP} = \{p, p', \dots p^{(k)}\}$.

- Define $\Phi = \{p_a : p \in \mathsf{PROP}, 1 \leq a \leq n\} = \{p_1, p'_1, \dots p_1^{(k)}, p_2, p'_2, \dots p_n^{(k)}\}$.
- For each $a \in A$ define $\Phi_a = \{p_a : p \in \mathsf{PROP}\} \subseteq \Phi$. It is easy to see that this will give a partition of Φ.
- For each $a \in A$, define a translation $\mathcal{T}_a : \mathcal{L}_{BNG} \to \mathcal{L}_{IBG}$ inductively as follows:

$$\begin{array}{lll} p^{\mathcal{T}_a} = p_a & (\varphi \vee \psi)^{\mathcal{T}_a} = \varphi^{\mathcal{T}_a} \vee \psi^{\mathcal{T}_a} & (X\varphi)^{\mathcal{T}_a} = \mathbf{X}\varphi^{\mathcal{T}_a} \\ (\neg\varphi)^{\mathcal{T}_a} = \neg\varphi^{\mathcal{T}_a} & (\Box\varphi)^{\mathcal{T}_a} = \displaystyle\bigwedge_{b \in R_a} \varphi^{\mathcal{T}_b} & (\varphi U \psi)^{\mathcal{T}_a} = \varphi^{\mathcal{T}_a}\mathbf{U}\psi^{\mathcal{T}_a} \end{array}$$

  where $p \in \mathsf{PROP}$. This translation accounts for the change to indexed propositions. Note that the replacement of $\Box\varphi$ with a conjunction indicating φ should be true at all the neighbours of a implicitly encodes R.
- Define $\mathcal{T}_g : A \to \mathcal{L}_{IBG}$ by $\mathcal{T}_g(a) = \bigwedge_{p \in g(a)} p^{\mathcal{T}_a} \wedge \bigwedge_{p \notin g(a)} \neg p^{\mathcal{T}_a}$. Recall that BNGs specify a start state (g) where IBGs do not. This function is used to encourage the start state to be met in the IBG.
- Define the iterated boolean game $\mathcal{T}(G)$ as

$$\mathcal{T}(G) = \left(A, \Phi, \Phi_1, \dots \Phi_n, \gamma(1)^{\mathcal{T}_1} \wedge \mathcal{T}_g(1), \dots \gamma(n)^{\mathcal{T}_n} \wedge \mathcal{T}_g(n)\right).$$

  Here each player has the translation of its BNG goal and also its required start state as its goal for $\mathcal{T}(G)$.

Thus we have the same agents in $\mathcal{T}(G)$ as in G. The set of propositional variables of $\mathcal{T}(G)$ is the set PROP, indexed by the agents in A. Each agent controls exactly those variables indexed by it, and so the sets Φ_a are all disjoint. Intuitively, we are using the sets Φ_a as the propositions at a's location in $\langle A, R \rangle$.

5.1.2 Strategy Translation

For strategy $s(a) = \langle N_{sa}, T_{sa}, I_{sa}, O_{sa} \rangle$ define

$$\mathcal{T}(s(a)) = (N_{sa}, I_{sa}, T_{sa}^{\mathcal{T}}, O_{sa}^{\mathcal{T}})$$

where $T_{sa}^{\mathcal{T}} : N_{sa} \times \mathcal{P}(\Phi) \to N_{sa}$ is defined by

$$T_{sa}^{\mathcal{T}}(v, V) = T_{sa}(v, \{\langle b, \{p \in \mathsf{PROP} \mid p^{\mathcal{T}_b} \in V\} \rangle \mid b \in R_a\})$$

(we ignore elements of V not in the neighbourhood of a and treat p_b as being p at b) and where $O_{sa}^{\mathcal{T}} : N_{sa} \to \mathcal{P}(\Phi_a)$ is defined as

$$O_{sa}^{\mathcal{T}}(v) = \{p^{\mathcal{T}_a} \mid p \in O_{sa}(v)\}.$$

So a strategy is translated by keeping the same nodes, using the same transition function (by restricting inputs to those acceptable for that function) and translating outputs.

We define the translation of the strategy profile s as

$$s^{\mathcal{T}} = (\mathcal{T}(s(1)), \dots \mathcal{T}(s(n))).$$

5.1.3 Properties of $\mathcal{T}$

We now consider which properties are preserved under $\mathcal{T}$. We specifically consider translations of games with strategies, as this allows us to consider questions of Nash equilibria. Due to limited space, some proofs have been omitted, but we have given brief description of them where possible.

We begin by showing that $\mathcal{T}(G)$ gives the same outcomes as G. This establishes that truth of formulas is preserved by $\mathcal{T}$.

Lemma 3 (Preservation of Outcomes)**.** *Let $G = \langle M, \gamma \rangle$ be a BNG where $M = \langle A, R, g \rangle$ and let s be a strategy profile for G. Then*

$$M, s, a, i \models \varphi \quad \textit{iff} \quad (\rho(s^{\mathcal{T}}), i) \models \varphi^{\mathcal{T}_a}$$

for every $a \in A$, $\varphi \in \mathcal{L}_{BNG}$ and timestep i, where the right hand side is taken with respect to $\mathcal{T}(G)$. In particular, $M, s, a \models \varphi$ iff $\rho(s^{\mathcal{T}}) \models \varphi^{\mathcal{T}_a}$.

Proof. By induction on the complexity of φ. The case $\varphi = p$ can be proved by an induction on i. The propositional and LTL cases are trivial. The case for $\Box\varphi$ remains.

$$\begin{aligned} M, s, a, i \models \Box\varphi \quad &\text{iff} \quad M, s, b, i \models \varphi \text{ for all } b \in R_a \\ &\text{iff} \quad (\rho(\mathcal{T}(s)), i) \models \varphi^{\mathcal{T}_b} \text{ for all } b \in R_a \text{ by inductive hypothesis} \\ &\text{iff} \quad (\rho(\mathcal{T}(s)), i) \models (\Box\varphi)^{\mathcal{T}_a}. \end{aligned}$$

For the particular result, recall that $M, s, a \models \varphi$ is defined as $M, s, a, 0 \models \varphi$ and $\rho(s^{\mathcal{T}}) \models \varphi^{\mathcal{T}_a}$ as $(\rho(s^{\mathcal{T}}), 0) \models \varphi^{\mathcal{T}_a}$.

We can conclude that the truth of all $\mathcal{L}_{BNG}$ formulas is preserved under $\mathcal{T}$, where formulas are translated relative to an agent. We have successfully simulated G as an IBG, using a BNG strategy.

The translation of formulas has been successful, so let us consider the translated goals. Recall that in $\mathcal{T}(G)$, agent a's goal is $\gamma(a) \wedge \mathcal{T}_g(a)$, where $\mathcal{T}_g(a)$ is the conjunction of the propositions in a's start state. If a obtains its goal in G with s, does it in $\mathcal{T}(G)$ with $\mathcal{T}(s)$? Using Lemma 3, this reduces to asking if $\rho(s^{\mathcal{T}}) \models \mathcal{T}_g(a)$ for every agent a. This can be shown by noting that a must use an available strategy. It follows that agents' utilities are preserved under $\mathcal{T}$. That is, a obtains its goal with s in G iff a obtains its goal with $\mathcal{T}(s)$ in $\mathcal{T}(G)$.

Let us now consider how Nash equilibria are affected by $\mathcal{T}$. If s is not a Nash equilibrium for G could $\mathcal{T}(s)$ be an Nash equilibrium for $\mathcal{T}(G)$? No. If s is not a Nash equilibrium, then some player a can do better with a different strategy $s_{a:♲}$. By Lemma 3, a can do better with $\mathcal{T}(♲)$ in $\mathcal{T}(G)$. Hence we have Lemma 4.

Lemma 4. *Let G be a BNG. Then $\mathcal{T}(\overline{NE(G)}) \subseteq \overline{NE(\mathcal{T}(G))}$.*

But what if s *is* a Nash equilibrium in G? Will $\mathcal{T}(s)$ be a Nash equilibrium of $\mathcal{T}(G)$? In order to answer this question, we will make use of myopic strategies, as introduced in Section 4.1.1. Specifically, we will use the myopic IBG strategies from [10], which are defined in a similar way to BNG myopic strategies. For myopic IBG strategies, we write the transition function as $\delta(q)$ since the valuation does not matter. By a very similar argument to Lemma 2, we can establish Lemma 5, its IBG parallel.

Lemma 5. *Let G be an IBG and $\vec{\sigma}$ be a strategy profile for G. Then for every player a there is a myopic strategy σ'_a such that $\rho(\vec{\sigma}) = \rho(\vec{\sigma}_{-a}, \sigma'_a)$.*

We are now ready to answer our question: if s is a Nash equilibrium for G, is $\mathcal{T}(s)$ a Nash equilibrium for $\mathcal{T}(G)$?

Lemma 6. *Let G be a BNG. Then $\mathcal{T}(NE(G)) \subseteq NE(\mathcal{T}(G))$.*

Proof. By contradiction. Suppose $s \in NE(G)$ but $s^{\mathcal{T}} \notin NE(\mathcal{T}(G))$. So there is a player a and a strategy $\sigma'_a = (Q'_a, q^{0'}_a, \delta'_a, \tau'_a)$ such that $\rho(s^{\mathcal{T}}) \not\models \gamma(a)^{\mathcal{T}_a} \wedge \mathcal{T}_g(a)$ and $\rho(s^{\mathcal{T}}_{-a}, \sigma'_a) \models \gamma(a)^{\mathcal{T}_a} \wedge \mathcal{T}_g(a)$. By Lemma 5 we can assume σ'_a is myopic.

Define a myopic BNG strategy $♲ = \langle Q'_a, T'_a, q^{0'}_a, O'_a\rangle$ for a such that $T'_a(v, g_a) = \delta'_a(v)$ and $O'_a(v) = \{p \in \mathsf{PROP} \mid p^{\mathcal{T}_a} \in \tau'_a(v)\}$. Now $\mathcal{T}(♲) = (Q'_a, q^{0'}_a, {T'_a}^{\mathcal{T}}, {O'_a}^{\mathcal{T}})$, where

$${T'_a}^{\mathcal{T}}(v, V) = T'_a(v, \{\langle b, \{p \in \mathsf{PROP} \mid p^{\mathcal{T}_b} \in V\}\rangle \mid b \in R_a\}) = \delta'_a(v)$$

$${O'_a}^{\mathcal{T}}(v) = \{p^{\mathcal{T}_a} \mid p \in O'_a(v)\} = \tau'_a(v).$$

So $\mathcal{T}(♲) = \sigma'_a$. We know $♲$ is available for a since $\rho(s^{\mathcal{T}}_{-a}, \sigma'_a) \models \mathcal{T}_g(a)$. It follows that $\rho({s_{a:♲}}^{\mathcal{T}}) = \rho(s^{\mathcal{T}}_{-a}, \sigma'_a)$. Since $\rho(s^{\mathcal{T}}_{-a}, \sigma'_a) \models \gamma(a)^{\mathcal{T}_a} \wedge \mathcal{T}_g(a)$ it must be that $M, s_{a:♲}, a \Vdash \gamma(a)$ by Lemma 3. Similarly, since $\rho(s^{\mathcal{T}}) \not\models \gamma(a)^{\mathcal{T}_a} \wedge \mathcal{T}_g(a)$ it must be that $M, s, a \nVdash \gamma(a)$. But then $u_a(s_{a:♲}) > u_a(s)$ so $s \notin NE(G)$, a contradiction. □

Thus Nash equilibria are preserved and non-Nash equilibria are preserved. We can summarise these results with the following theorem.

Theorem 1. *Let G be a BNG. Then $s \in NE(G)$ iff $\mathcal{T}(s) \in NE(\mathcal{T}(G))$.*

Proof. Left to right is by Lemma 6. Right to left is by contrapositive, using Lemma 4. □

The reader should take care to note that Theorem 1 does *not* say $\mathcal{T}(NE(G)) = NE(\mathcal{T}(G))$. Indeed, if $s \in NE(G)$ and there are players a, b with $b \notin R_a$ then $s^{\mathcal{T}}_a$ has no transitions depending on the state of b. So we can modify $s^{\mathcal{T}}_a$ to $s^{\mathcal{T}'}_a$ by duplicating some node, and modifying the transition function so that it goes to a different duplicate depending on the state of b. We still have $s^{\mathcal{T}'}_a \in NE(\mathcal{T}(G))$ so in this case $\mathcal{T}(NE(G)) \subsetneq NE(\mathcal{T}(G))$.

The reduction from BNGs to IBGs considered here is similar to that given from Epistemic Boolean Games to Boolean Games in [8]. Specifically, both translations involve a possibly exponential increase in the size of goal formulas. As is noted in [8], unless $P = PSPACE$ (considered highly unlikely), this means BNGs have an exponential increase in succinctness over IBGs.

5.1.4 Example: Unstable Colouring

Recall the unstable diversity game G_{3CU} from Section 3.2.1 (depicted in Figure 5.1). We will take the same strategy profile s that we used then, with players a and b using TaFT and player c using TiFT. As we established, s is a Nash equilibrium for G_{3CU}.

In order to translate G, let us define it more rigourously. $G = \langle M, \gamma\rangle$. We have $M = \langle A, R, g\rangle$ where $A = \{a, b, c\}$, $R = \{\langle a, b\rangle, \langle b, c\rangle, \langle c, a\rangle\}$ and $g(x) = \{p\}$ for all $x \in A$. Further, $\gamma(x) = GF\text{Div}$ for all $x \in A$.

The set of players remains unchanged in $\mathcal{T}(G)$, so we keep $A = \{a, b, c\}$. For the finite set of variables we define $\Phi = \{p_a, p_b, p_c\}$, indexing the propositional variable p by each agent. Φ is partitioned to give the variables each agent controls. This is straightforward:

$$\Phi = \{p_a, p_b, p_c\} \qquad \Phi_a = \{p_a\} \qquad \Phi_b = \{p_b\} \qquad \Phi_c = \{p_c\}$$

Finally we must define each player's goal. For these, we translate the original goal and specify the start state. Recall that $\mathsf{Div} = (p \wedge \Box\neg p) \vee (\neg p \wedge \Box p)$. From our translation, we have that

$$(GF\mathsf{Div})^{\mathcal{T}_a} = \mathbf{GF}(((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b))),$$

with similar results for b and c. The IBG goal for each player includes their start state, hence we have that

$$\mathcal{T}(G) = (A, \Phi, \Phi_a, \Phi_b, \Phi_c, (GF\mathsf{Div})^{\mathcal{T}_a} \wedge p_a, (GF\mathsf{Div})^{\mathcal{T}_b} \wedge p_b, (GF\mathsf{Div})^{\mathcal{T}_c} \wedge p_c).$$

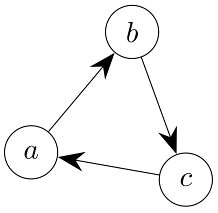
Now consider the translation $s^{\mathcal{T}}$ of the strategy profile s. We will detail the translation of TaFT for a, and then present all three translated strategies.

The set of nodes for TaFT is $\{q_0, q_1\}$, and this is preserved in the translation. For the transition function, we must now take into account the behaviour of every player. We are only concerned with what b is doing, so we disregard the behaviour of other players. The outputs are simply the translation of the TaFT outputs. The transition from q_0 to q_1 is marked with two possible inputs, $\{p_a, p_b\}$ and $\{p_a, p_b, p_c\}$. In TaFT, a transitions from q_0 exactly when b has p. The behaviour of c is ignored, as it is here. Note that, in keeping with the notation used in [10], we have excluded impossible transitions, such as $\emptyset$ from q_0 (since a controls p_a and it has been set true, there is no way for this valuation to occur at q_0). Figure 5.2 gives all three translated strategies.

Next we will establish that $s^{\mathcal{T}}$ is a Nash equilibrium for $\mathcal{T}(G)$. In the run, every player starts by choosing true for their variable and so the initial state is $\{p_a, p_b, p_c\}$. Given the transitions, we get the following run.

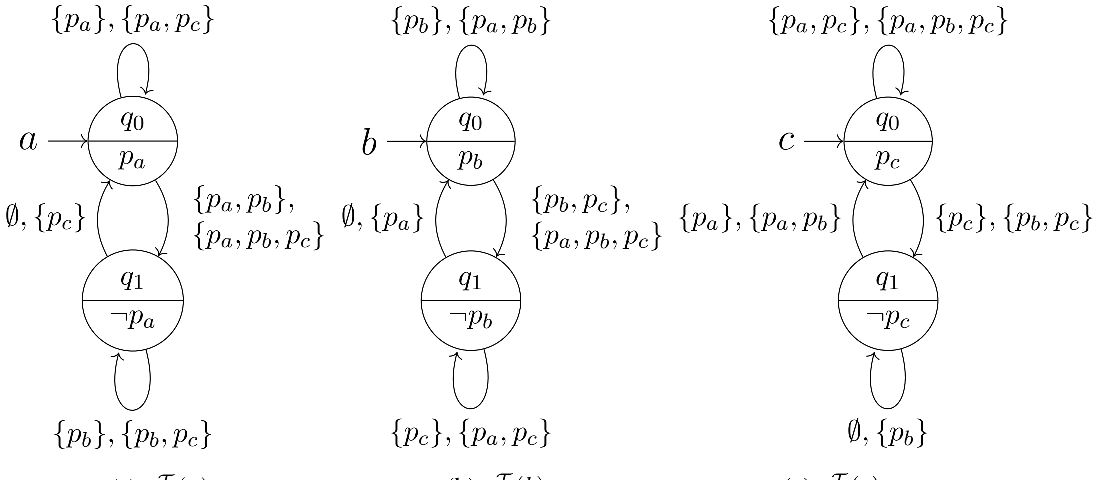
$$\{p_a, p_b, p_c\}, \{p_c\}, \{p_a\}, \{p_a, p_b, p_c\}, \{p_c\}, \{p_a\}, \ldots$$

Recall that a's goal is $\mathbf{GF}((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)) \wedge p_a$. Clearly p_a is satisfied. Every third timestep we have $p_a \wedge \neg p_b$, so $\mathbf{GF}((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b))$ is satisfied and hence a's goal is. b achieves its goal similarly, since every second timestep satisfies $\neg p_b \wedge p_c$. This second timestep also gives c its goal, since it satisfies $p_c \wedge \neg p_a$. Hence every player achieves its goal, and so the Nash Equilibrium is preserved.



$$\gamma(a) = GF\mathsf{Div} \qquad g(a) = \{p\}$$
$$\gamma(b) = GF\mathsf{Div} \qquad g(b) = \{p\}$$
$$\gamma(c) = GF\mathsf{Div} \qquad g(c) = \{p\}$$

Figure 5.1: G_{3CU}. 3-cycle unstable colouring game.



(a) $s^{\mathcal{T}}(a)$ (b) $s^{\mathcal{T}}(b)$ (c) $s^{\mathcal{T}}(c)$

Figure 5.2: Strategy profile $s^{\mathcal{T}}$.

5.2 Translation from Iterated Boolean Games to Boolean Network Games

Now we consider the opposite direction, simulating an IBG as a BNG. Again, by abuse of notation, we use T to represent any functions used in the translation.

5.2.1 Game Translation

Suppose $G = (A, \Phi, \Phi_1, \dots \Phi_n, \gamma_1, \dots \gamma_n)$ is an IBG. Set $R = \{\langle a, b\rangle \mid a \neq b\}$. Thus every player can see every other player. Use Φ for PROP. Define a translation $\mathsf{T}_a : \mathcal{L}_{IBG} \to \mathcal{L}_{BNG}$ for each $a \in A$ as follows:

$$p^{\mathsf{T}_a} = \begin{cases} p & \text{if } p \in \Phi_a \\ \Diamond p & \text{if } p \notin \Phi_a \end{cases} \qquad (\neg\varphi)^{\mathsf{T}_a} = \neg\varphi^{\mathsf{T}_a}$$
$$(\varphi \vee \psi)^{\mathsf{T}_a} = \varphi^{\mathsf{T}_a} \vee \psi^{\mathsf{T}_a}$$
$$(\mathbf{X}\varphi)^{\mathsf{T}_a} = X\varphi^{\mathsf{T}_a} \qquad (\varphi\mathbf{U}\psi)^{\mathsf{T}_a} = \varphi^{\mathsf{T}_a}U\psi^{\mathsf{T}_a}.$$

Define a goal profile γ as

$$\gamma(a) = \gamma_a^{\mathsf{T}_a}.$$

Since BNGs require a specified initial state, and IBGs do not, we finish defining $\mathsf{T}(G)$ once we have defined translation for strategies.

5.2.2 Strategy Translation

Consider the strategy $\sigma_a = (Q_a, q_a^0, \delta_a, \tau_a)$.

In the translated game, each player has control over all the variables in $\mathsf{PROP} = \Phi$, including those they do not control in the IBG. In the translation, each player's strategy sets all the variables they "should not control" to false. That is,

$$\tau_a^{\mathsf{T}}(q_a^k) = \tau_a(q_a^k).$$

This explains our translation of formulas T_a. If player a wants p in the IBG, then they want the player controlling p to set it true. So if a controls p, then in the translation a wants p. If a does not control p, they want the player who controls p to set it true. Every player who does not control p will set it false, so a wants $\Diamond p$.

The transition function only considers the value of the variables at the players who "should be" controlling them. So we should evaluate using only values from correct players. Hence define the translation of δ_a as

$$\delta_a^{\mathsf{T}}(q_a^k, g_a) = \delta_a\left(q_a^k, \bigcup_{b\in A} g(b) \cap \Phi_b\right).$$

We can now define the translation of σ_a.

$$\mathsf{T}(\sigma_a) = \langle Q_a, \delta_a^{\mathsf{T}}, q_a^0, \tau_a^{\mathsf{T}}\rangle.$$

If $\vec{\sigma} = (\sigma_1, \dots \sigma_n)$ is a strategy profile, define $\vec{\sigma}^{\mathsf{T}}$ such that $\vec{\sigma}^{\mathsf{T}}(a) = \mathsf{T}(\sigma_a)$.

5.2.3 Game Translation (continued)

Take G from above and a strategy profile $\vec{\sigma} = (\sigma_1, \dots \sigma_n)$, where $\sigma_a = (Q_a, q_a^0, \delta_a, \tau_a)$ for all a. Define a global state $g_{\vec{\sigma}}$ where

$$g_{\vec{\sigma}}(a) = \tau_a(q_a^0).$$

That is, each player's initial state is the initial state of its strategy.

Now take $\mathsf{T}(G, \vec{\sigma}) = \langle\langle A, R, g_{\vec{\sigma}}\rangle, \gamma\rangle$, where R, γ are defined as above. This gives us a BNG corresponding to both G, with start state corresponding to $\vec{\sigma}$. We write $\mathsf{T}(G)$ for the set of possible translations of G, and also when it is clear which strategy is being used for translation.

5.2.4 Properties of T

We now consider properties of T. Our goal is to show similar properties to those we proved for $\mathcal{T}$, namely that game outcomes, player utilities and Nash equilibria are all preserved under T.

The parallel of Lemma 3 becomes trickier for T since p's translation depends on which agent we are evaluating at. To help the proof, we have the following lemma.

Lemma 7. *Given an IBG G, a strategy profile $\vec{\sigma}$ for G and a timestep i. For every agent a and $p \in \Phi_a$, we have $p \in \rho(\vec{\sigma})[i]$ iff $p \in g^i(a)$, where g^i is the corresponding global state in the translation.*

Proof. By induction on i. $\square$

We build on this result to show that truth of formulas is preserved under T.

Lemma 8 (Preservation of Outcomes)**.** *Let $G = (A, \Phi, \Phi_1, \dots \Phi_n, \gamma_1, \dots \gamma_n)$ be an IBG and $\vec{\sigma}$ a strategy profile for G. Suppose $\mathsf{T}(G, \vec{\sigma}) = \langle M_{\mathsf{T}}, \gamma_{\mathsf{T}}\rangle$ Then*

$$(\rho(\vec{\sigma}), i) \models \varphi \quad \textit{iff} \quad M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}, a, i \models \varphi^{\mathsf{T}_a}$$

for every $a \in A$, formula $\varphi \in \mathcal{L}_{IBG}$ and timestep i.

Proof. By induction on the complexity of φ. For the case $\varphi = p$ there are two subcases. If $p \in \Phi_a$ then the case follows from Lemma 7. So suppose $p \notin \Phi_a$ and $(\rho(\vec{\sigma}), i) \vDash p$. There is a $b \in A$ such that $p \in \Phi_b$ since the agent-indexed sets partition Φ. By Lemma 7, $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}, b, i \vDash p$. By the structure of $\mathsf{T}(G)$, Rab and so we have $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}, b, i \vDash \Diamond p$. So $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}, b, i \vDash p^{\mathsf{T}_a}$. The other direction is similar. The cases when $\varphi \neq p$ are routine. ∎

We have established that truth of formulas is preserved under translation. Since players' goals in $\mathsf{T}(G)$ are simply translations of their goals in G it follows that players' utilities are preserved under translation.

Let us now consider Nash equilibria. First, if $\vec{\sigma}$ is not a Nash equilibrium for G, can we be sure that $\mathsf{T}(\vec{\sigma})$ is not a Nash equilibrium for $\mathsf{T}(G)$? Yes. As with Lemma 4, if a can do better by changing its strategy to σ'_a in G, then a can do better by changing its strategy to $\mathsf{T}(\sigma'_a)$ in $\mathsf{T}(G)$.

Lemma 9. *Let G be an IBG. Then* $\mathsf{T}(\overline{NE(G)}) \subseteq \overline{NE(\mathsf{T}(G))}$.

Non-Nash equilibria are preserved, but what about Nash equilibria? We will make use of myopic strategies once again, and us Lemma 2, the existence of equivalent myopic strategies. In the parallel argument for $\mathcal{T}$, we next showed that $\mathcal{T}(NE(G)) \subseteq NE(\mathcal{T}(G))$ (Lemma 6). The proof hinged on our being able to find a BNG strategy ♻ to map onto the myopic σ'_a. This was a straightforward exercise. If we attempt the same for T, we reach a problem: we do not know that the myopic strategy only outputs allowable valuations. Since players in $\mathsf{T}(G)$ have control over all propositions, perhaps a's better strategy involves changing a proposition it can't change in G. In order to account for this problem, we provide the following lemma.

Lemma 10. *Let G be an IBG and $\vec{\sigma}$ a strategy profile for G. Then for every BNG strategy* ♻ $= \langle N, T, I, O\rangle$ *for a there is an IBG strategy σ'_a for a such that*

$$M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:♻}, a, i \vDash \varphi^{\mathsf{T}_a} \quad \textit{iff} \quad M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a, i \vDash \varphi^{\mathsf{T}_a}$$

for every timestep i.

Proof. By Lemma 2 we can assume ♻ is myopic. Define $\sigma'_a = (N, I, T', O')$ where $T'(q, \vec{v}) = T(q)$ for all $\vec{v}$ and $O'(q) = O(q) \cap \Phi_a$ So σ'_a is ♻ with outputs restricted to Φ_a. We claim that σ'_a satisfies the requirements of the lemma. Let the outcome behaviour of M_{T} with $\vec{\sigma}^{\mathsf{T}}_{a:♻'}$ be $\{\langle g^i, \xi^i\rangle\}_{i\in\mathbb{N}}$ and the outcome behaviour of M'_{T} with $(\vec{\sigma}_{-a}, \sigma_a)^{\mathsf{T}}$ be $\{\langle g'^i, \xi'^i\rangle\}_{i\in\mathbb{N}}$. An induction on i establishes that

$$g^i(a) \cap \Phi_a = g'^i(a) \qquad\qquad \xi^i(a) = \xi'^i(a) \tag{5.1}$$

$$g^i(b) = g'^i(b) \qquad\qquad \xi^i(b) = \xi'^i(b) \tag{5.2}$$

for every $b \in A \setminus \{a\}$ and timestep i. We now proceed by induction on the complexity of φ to show that

$$M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:♻}, a, i \vDash \varphi^{\mathsf{T}_a} \quad \text{iff} \quad M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a, i \vDash \varphi^{\mathsf{T}_a}.$$

First suppose $\varphi = p$. If $p \in \Phi_a$ then $p^{\mathsf{T}_a} = p$ and $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:♻}, a, i \vDash p$ iff $p \in g^i(a)$. By 5.1, this is the case iff $p \in g'^i(a)$ since $p \in \Phi_a$. But this means $M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a, i \vDash p$. If $p \notin \Phi_a$

then $p^{\mathsf{T}_a} = \Diamond p$. Again we have $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:\text{♲}}, a, i \vDash \Diamond p$ iff $p \in g^i(b)$ for some b with Rab. By 5.2 this means $p \in g'^i(b)$ and so $M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a, i \vDash \Diamond p$.

The propositional and temporal cases follow by routine arguments. Thus σ'_a fulfils our requirements and we have our result.

With Lemma 10 proved it is now straightforward to obtain the parallel of Lemma 6.

Lemma 11. *Let G be an IBG. Then* $\mathsf{T}(NE(G)) \subseteq NE(\mathsf{T}(G))$.

Proof. Suppose $\vec{\sigma} \in NE(G)$. Suppose for contradiction that $\vec{\sigma}^{\mathsf{T}} \notin NE(\mathsf{T}(G))$. Then there is a player $a \in A$ and a BNG strategy ♲ for a such that $u_a(\vec{\sigma}^{\mathsf{T}}_{a:\text{♲}}) > u_a(\vec{\sigma}^{\mathsf{T}})$.

We must have $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:\text{♲}}, a \vDash \gamma(a)$ and $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}, a \nvDash \gamma(a)$ by the definition of u_a. Since $\gamma(a) = {\gamma_a}^{\mathsf{T}_a}$, by Lemma 8 (Preservation of Outcomes) we have $\rho(\vec{\sigma}) \nvDash \gamma_a$. By Lemma 10 there is a strategy σ'_a for a such that

$$M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:\text{♲}}, a \vDash \varphi^{\mathsf{T}_a} \quad \text{iff} \qquad M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a \vDash \varphi^{\mathsf{T}_a}.$$

Since $M_{\mathsf{T}}, \vec{\sigma}^{\mathsf{T}}_{a:\text{♲}}, a \vDash \gamma(a)$ we have $M'_{\mathsf{T}}, (\vec{\sigma}_{-a}, \sigma'_a)^{\mathsf{T}}, a \vDash \gamma(a)$. So by Lemma 8, $\rho(\vec{\sigma}_{-a}, \sigma'_a) \vDash \gamma_a$.

But now we have $\rho(\vec{\sigma}_{-a}, \sigma'_a) \vDash \gamma_a$ and $\rho(\vec{\sigma}) \nvDash \gamma_a$, so $\rho(\vec{\sigma}) \not\succsim_a \rho(\vec{\sigma}_{-a}, \sigma'_a)$ and hence $\vec{\sigma} \notin NE(G)$, a contradiction.

Finally we are ready to prove the parallel of Theorem 1.

Theorem 2. *Let G be an IBG. Then* $\vec{\sigma} \in NE(G)$ *iff* $\mathsf{T}(\vec{\sigma}) \in NE(\mathsf{T}(G))$.

Proof. Left to right is by Lemma 11. Right to left is by contrapositive, using Lemma 9.

As with Theorem 1, we note that this does not mean $\mathsf{T}(NE(G)) = NE(\mathsf{T}(G))$ and indeed basic changes to strategies can show this is the case.

It is crucial to realise that T is not an inverse of $\mathcal{T}$. Indeed, it is the case that $\mathsf{T}(\mathcal{T}(G)) \neq G$ and $\mathcal{T}(\mathsf{T}(G)) \neq G$. Since $\mathcal{T}$ indexes propositions by agent and T allows all agents control over every proposition, iterated applications of $\mathcal{T}$ and T will increase the number of propositions in the game.

Some say Surak crossed this expanse when the hot blood of battle still flowed green, but with logic, he cooled it.

Arev, *The Forge*,
STAR TREK: ENTERPRISE

Chapter 6

Modal Strategies and Logical Representation of Strategy Profiles

In Iterated Boolean Games [10], the authors demonstrate a logical characterisation of strategies which allows for strategy profiles to be represented as single LTL formulas. Any LTL run in which this formula is satisfied matches the run of the IBG with that strategy profile.

It would be nice to have a similar result for Boolean Network Games. Unfortunately, as we will discuss, this is not possible in all cases. The transition function for strategies is more powerful than the language can handle, and so we cannot represent all strategy profiles in $\mathcal{L}_{BNG}$. Further, since players control the same propositions, it becomes difficult to enforce different strategies on players, in the way [10] do. We can, however, represent a subset of strategies, as we will see.

6.1 Restricting Strategies

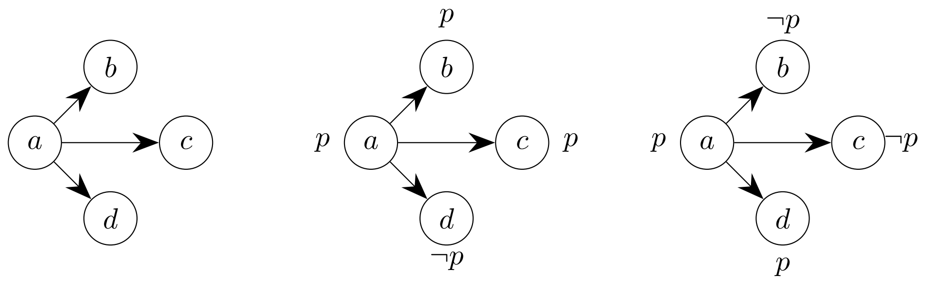
First we will establish that the transition function of BNGs is too powerful for modally representable strategies. In this section we will give an example of this power, and provide a suitable restriction of strategies to *modal* strategies.

6.1.1 Modal Strategies

Consider network N described in Figure 6.1a, and the strategies a_1 and a_2 for player a described in Figure 6.2. Recall that by the definition of a strategy (Section 3.1.2), the transition function can differentiate between neighbours, so these are valid strategies. In strategy a_1, a starts in q_0 and changes state whenever b, c, d all choose p. In a_2, a requires that b and c choose p, and that d does not, in order to change state. For the purposes of our discussion, the players' goals, and b, c, d's strategies, are unimportant.

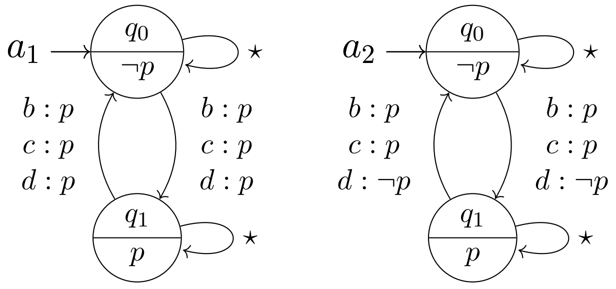
Let us examine the conditions for changing state in a_1 and a_2. The transitions in a_1 are intuitively captured by requiring $\Box p$ to hold. So we could logically represent a_1 by something of the form $((\pm p \wedge \Box p) \rightarrow X \mp p)$.

However no modal formula captures the transition in a_2. This can be shown by the two models M, M' in Figures 6.1b and 6.1c, respectively. In M_1 exactly a, b, c have p and in M_2 exactly a, d have p in M_2. So M_1 satisfies the transition in a_2 but M_2 does not. It is easy



(a) Network N. (b) Model M_1. (c) Model M_2.

Figure 6.1: The network N and two models based on it.



(a) Strategy a_1. (b) Strategy a_2.

Figure 6.2: Non-modal strategies.

to see that M_1 and M_2 are *bisimilar* at a. Specifically, $M_1, a \leftrightarroweq_1 M_2, a$ (where $\leftrightarroweq_1$ indicates bisimulation to depth 1). Hence by standard results (see, for example, [4]) M_1 is modally indistinguishable from M_2 by a. So no formula will allow us to transition in M_1 without also transitioning in M_2. Thus, no formula can represent the transition in a_2.

If we are to logically represent strategies by sets of modal formulas we must isolate those strategies in which this can be achieved. That is, we wish to capture strategies like a_1, where transitions can be encoded by formulas, while ruling out those like a_2, with transitions more powerful than that which our formulas can represent. To this purpose, we take the following definition.

Definition 5 (Bisimilar Environments). *Two environments $e_1 : R_a \to \mathcal{P}(\mathsf{PROP})$ and $e_2 : R_a \to \mathcal{P}(\mathsf{PROP})$ are* bisimilar *for a (denoted $e_1 \leftrightarroweq_a e_2$) if there are two models M_1, M_2, satisfying e_1 and e_2 respectively, such that $M_1, a \leftrightarroweq_1 M_2, a$.*

Intuitively, $e_1 \leftrightarroweq_a e_2$ means that if a has a neighbour with p in e_1 then a does in e_2 too, and if a has a neighbour with $\neg p$ in e_1 then similarly in e_2, and vice versa. Now we can give the definition of a modal strategy.

Definition 6 (Modal Strategy). *A modal strategy for a is a strategy $\langle N, T, I, O \rangle$ for a with the requirement that for all $q \in N$ and environments e_1, e_2 of a, $T(q, e_1) = T(q, e_2)$ whenever $e_1 \leftrightarroweq_a e_2$.*

The intuition for this is that the function T can only determine the existence of different subsets of PROP. It is clear that the set of modal strategies is a proper subset of the set of strategies - a_1 is a strategy but a_2 is not modal.

If $\langle N, T, I, O\rangle$, then we can abbreviate the transition function by using modal formulas. That is, we can write $T(q, \Box p)$ to indicate the transition from q when every neighbour has p. Similarly, $T(q, \Diamond p \wedge \Diamond\neg p)$ indicates the transition when at least one neighbour has p and at least one neighbour does not.

6.1.2 Modal Strategies and Validity

Suppose we have a BNG G and a player a. Let s be a strategy profile for G such that $s(a) = \langle N, T, I, O\rangle$ is modal. For simplicity, suppose that $\mathsf{PROP} = \{p\}$.

Example 1. *Suppose there is exactly one $q \in N$ such that $O(q) = \{p\}$. We know that every other node in N has $\emptyset$ as its output. There are 3 possible transitions from q, namely $T(q, \Box p)$, $T(q, \Diamond p \wedge \Diamond\neg p)$ and $T(q, \Box\neg p)$. Suppose that $T(q, \Box p) = q$ and $T(q, \Diamond p \wedge \Diamond\neg p) \neq q \neq T(q, \Box\neg p)$. So only if a sees $\Box p$ will we stay in q.*

Then we have that

- $M, s, a \vDash G((p \wedge \Box p) \to Xp)$;
- $M, s, a \vDash G((p \wedge \Diamond p \wedge \Diamond\neg p) \to X\neg p)$;
- $M, s, a \vDash G((p \wedge \Box\neg p) \to X\neg p)$.

We have made many assumptions, but the underlying motivation should be clear. By adopting a modal strategy, we have made certain formulas true at a. The obvious question is of characterisation. That is, can we find a set of modal formulas which uniquely characterise each modal strategy? If so, then modal strategies may be able to be embedded into the model itself.

Example 2. *Consider TiFT$_U$, the unstable version of TIT-FOR-TAT described in Figure 6.3. If all our neighbours have the same colour, we change to that colour. If we have no neighbours we alternate indefinitely. Clearly this is modal. If $s(a)$ is this strategy, then we will have*

- $M, s, a \vDash G((\Box p) \to Xp)$;
- $M, s, a \vDash G((\Box\neg p) \to X\neg p)$;
- $M, s, a \vDash G((p \wedge \Diamond p \wedge \Diamond\neg p) \to Xp)$;
- $M, s, a \vDash G((\neg p \wedge \Diamond p \wedge \Diamond\neg p) \to X\neg p)$.

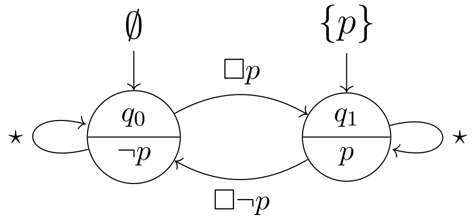


Figure 6.3: The Unstable TIT-FOR-TAT Strategy (TiFT$_U$)

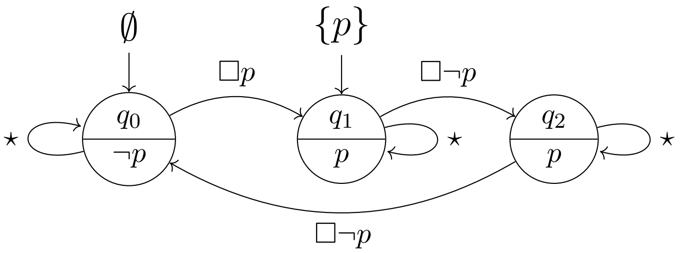


Figure 6.4: The strategy TIT-FOR-FOR-TAT (TiFFT).

The first two formulas encode the transitions for $\Box p$ and $\Box\neg p$. Notice that we do not need to specify which node we're in - whatever the case, after seeing $\Box p$ we will output p. The second two formulas are a result of the $\star$ transitions - if we see something that isn't a $\Box$ we should stay with the same value.

$$T = \left\{ \begin{array}{l} G((\Box p) \to Xp), G((\Box\neg p) \to X\neg p), \\ G((p \wedge \Diamond p \wedge \Diamond\neg p) \to Xp), G((\neg p \wedge \Diamond p \wedge \Diamond\neg p) \to X\neg p) \end{array} \right\}$$

T is the modal representation of TiFT$_U$. If an agent employs TiFT$_U$ then they will satisfy all formulas of G, and if an agent satisfies all formulas of G then they are employing a strategy (game) equivalent to TiFT$_U$.

We have established that TiFT$_U$ can be characterised by a set of modal formulas. Unfortunately, the existence of a modal characterisation becomes less clear if we make a minor alternation. Let us modify TiFT$_U$ to get the strategy TIT-FOR-FOR-TAT (TiFFT), described in Figure 6.4. In TiFFT, players stay in p for *two* iterations of their neighbours having p before copying them. This can be seen as the player giving two chances before giving up on p. Again, if a player has no neighbours they will keep changing states.

No obvious set of modal formulas representing TiFFT presents itself. We cannot use T from above, due to the state q_2. If we introduce modal operators to consider past events, we may have some chance though. Take S to be a *since* operator, where $\varphi S \psi$ means that φ has been true at every timestep since ψ was true (except possibly now). This operator was first introduced by Kamp [13], and corresponds to the past facing version of "until", U. Consider now the set T'.

$$T' = \left\{ \begin{array}{l} G((\neg p \wedge \Box p) \to Xp), G((\neg p \wedge \neg\Box p) \to X\neg p), \\ G((p \wedge \Diamond p) \to Xp), \\ G((p \wedge \Box\neg p \wedge (\Diamond p) S (\Box p)) \to Xp), \\ G((p \wedge \Box\neg p \wedge \neg(\Diamond p) S (\Box p)) \to X\neg p) \end{array} \right\}$$

The first two formulas accounts for q_0 in much the way q_0 is accounted for in T. The second formula says that if we're in a p-state and we see $\Diamond p$ then we will remain in a p-state. This is true in our transitions, even though we haven't differentiated which p-state we remain in. The differentiation comes in the final two formulas.

If we are in q_1 then we will have p. Further (assuming the game has been played for a while, and we've moved around the strategy), we haven't seen $\Box\neg p$ since we saw $\Box p$, so we have $(\neg\Box\neg p) S (\Box p)$. And if we see $\Box\neg p$ then we should move to q_1 and so output p at the next timestep. If we're in q_2 we will also have p, but we *will* have seen $\Diamond p$ since we saw $\neg p$. So next we should go into $\neg p$ (q_0).

This construction T' is not correct. Specifically, if we are at the beginning of the run, and q_0 has never been obtained, the antecedent of $((p \wedge \Box\neg p \wedge (\Diamond p)S(\neg p)) \to Xp)$ is always false. Even so, this example demonstrates that using temporal operators carefully may allow us to represent a greater range of strategies.

Even if possible, this construction seems messy. It would be nice to include more explicit references to which state we're in, to give a more intuitive set T. Indeed, this is how [10] proceeds and what we will now consider.

6.2 Using Secret States in Strategy Representation

In order to give a more intuitive representation of modal strategies as collections of modal formulas, we will introduce the concept of a *secret state*. This is a state that a player can be in, visible only to themselves, not any other players. We will use secret states to represent the node a player's strategy is in. This construction is very similar to the logical characterisation of machine strategies in [10].

6.2.1 Secret States

We take a (finite) set MOOD of propositional variables, disjoint from PROP, that are used to represent internal, secret states of each player. The transition function for strategies still takes into account only the external (PROP) states of neighbours, but goals are able to make use of internal states.

We can label internal states (subsets of MOOD) by $q_0, q_1, \ldots$, and take these to represent formulas true if, and only if, a player is in that state (this is possible since MOOD is finite). Clearly we are drawing a parallel here between internal states and the node a player's strategy is in. Let's make this explicit.

6.2.2 Secret State Strategies

Take again the strategy TIT-FOR-FOR-TAT from above. If we are allowed to use secret states, then giving a modal representation of it becomes almost trivial:

$$T'' = \left\{ \begin{array}{l} G((q_0 \wedge \Box p) \to Xq_1), G((q_0 \wedge \neg\Box p) \to Xq_0), \\ G((q_1 \wedge \Box\neg p) \to Xq_2), G((q_0 \wedge \neg\Box\neg p) \to Xq_1), \\ G((q_2 \wedge \Box\neg p) \to Xq_0), G((q_2 \wedge \neg\Box\neg p) \to Xq_2), \\ G(q_0 \to \neg p), G(q_1 \to p), G(q_2 \to p), p \to q_1, \neg p \to q_0 \end{array} \right\}.$$

Each transition can be encoded in a methodical manner. If we are in q_i and we see environment e then in the next step we go to q_j. For each node q_i the next 3 formulas specify the output of that node. The initial state requirements are given by the final 2 non-temporal formulas.

Secret state strategies provide a very intuitive way to capture machine strategies in $\mathcal{L}_{BNG}$. Unfortunately they are still limited in cases where different players have different strategies. In an IBG, each player controls a different set of propositional variables, so each strategy profile is able to be given a different logical representation, as was shown in [10].

It would be an interesting project to consider whether a similar representation can be found for every (modal) BNG strategy profile. This seems unlikely, since every player controls the same set of propositions and players could have radically different and contradictory strategies. It would be intriguing to see to what extent strategy profiles can be captured however.

If all objects are given, then at the same time all *possible* states of affairs are also given.

Tractatus Logico-Philosophicus, LUDWIG WITTGENSTEIN

Chapter 7

Epistemic Boolean Network Games

In this chapter we introduce an extension of Boolean Network Games to deal with the epistemic states of players. The motivation is for players to have goals relating to their knowledge, and the knowledge of other players. Perhaps a player a wishes to know what strategies their neighbours are using, but for their neighbours not to know that a knows this. This sort of goal can tie in with the modal strategies and logical representations from Chapter 6 (to know a player's strategy could be identified with knowing the set of formulas corresponding to their strategy holds). Our model is similar to the Epistemic Boolean Games from [8], with additional network structure and iteration.

Our presentation of these *Epistemic Boolean Network Games* (EBNGs) is intended as an investigation into how epistemic states could be implemented in BNGs. In Section 7.1 we introduce the structure on which our EBNGs will be played. Section 7.2 introduces the games proper, and discusses examples and potential modifications.

7.1 Epistemic Iterated Networks

Boolean Network Games were defined in two dimensions - a network structure, giving players' locations, and a temporal dimension. We are adding a third, epistemic dimension. For this, we introduce a new model.

An *epistemic iterated network* is a tuple $\langle A, R, W, V \rangle$ where A is a set of *agents*, $R \subseteq A^2$ is an accessibility relation between these agents, W is a set of epistemic possibilities and $V : W \times A \times \mathbb{N} \to \mathcal{P}(\mathsf{PROP})$ is a valuation function. R is the same relation as in BNGs, with R_a defined as a's neighbourhood like before.

The valuation function V assigns a set of propositions to every agent at every timestep and in every epistemic possibility. Since the model evolves over time, V tells us which agent chooses which state and when. Unlike BNGs, we have explicitly included the timestep in the model, rather than using a model update strategy. This is to allow for the potential of backward looking operators (as discussed briefly in Section 6.1.2). It also eases in the next definition.

We define an equivalence relation $\sim_a^i$ over W for each agent $a \in A$ and timestep $i \in \mathbb{N}$ by

$$w \sim_a^i v \quad \text{iff} \quad V(w, b, i) = V(v, b, i) \text{ for every } b \in R_a.$$

Essentially, w and v are equivalent to a if they do not differ from a's (depth 1) perspective.

From this, we define the relations $\approx_a^i$ inductively as

$$\begin{array}{rcl} \approx_a^0 & = & \sim_a^0 \\ \approx_a^{i+1} & = & \approx_a^i \cap \sim_a^i . \end{array}$$

If $w \approx_a^i v$ then a can't tell the difference between w and v, and never has been able to.

7.1.1 Language

We use the language of BNGs enriched with the epistemic operator K "knows".

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \Box\varphi \mid X\varphi \mid \varphi U \varphi \mid K\varphi$$

Formulas are evaluated with respect to an agent $a \in A$, a timestep i and an epistemic state w as follows:

$$\begin{array}{lll} M, V, w, a, i \models p & \text{iff} & p \in V(w, a, i) \\ M, V, w, a, i \models \neg\varphi & \text{iff} & M, V, w, a, i \not\models \varphi \\ M, V, w, a, i \models (\varphi \vee \psi) & \text{iff} & M, V, w, a, i \models \varphi \text{ or } M, s, w, a, i \models \psi \\ M, V, w, a, i \models \Box\varphi & \text{iff} & M, V, w, b, i \models \varphi \text{ for all } b \in R_a \\ M, V, w, a, i \models X\varphi & \text{iff} & M, V, w, a, i+1 \models \varphi \\ M, V, w, a, i \models \varphi U \psi & \text{iff} & M, V, w, a, k \models \psi \text{ for some } i \leq k \\ & & \text{and } M, V, w, a, j \models \varphi \text{ for all } i \leq j < k. \\ M, V, w, a, i \models K\varphi & \text{iff} & M, V, v, a, i \models \varphi \text{ for all } v \approx_a^i w \end{array}$$

As before, we write $M, V, w, a \models \varphi$ iff $M, V, w, a, 0 \models \varphi$.

Note that K is evaluated at a player. Players' goals are able to reason about other players' knowledge, but only using modalities. That is, we could have the goal $\Box K\varphi$, but not the goal that player c has $K\varphi$. Since K is defined with respect to an equivalence relation, it is an $S5$ modality.

7.2 Epistemic Boolean Network Games

We have a network for EBNGS, so now we can introduce the games themselves. Strategies are defined as with normal BNGs. Given a strategy profile s and an epistemic iterated network $\langle A, R, W, V\rangle$, we say $w \in W$ *satisfies* s if the following condition holds:

- $M, V, w, a, i \models p$ iff $p \in g^i(a)$, for every agent $a \in A$ and $i \in \mathbb{N}$.

That is, the projection of the network's evolution onto w gives the run of the corresponding BNG with strategy profile s.

An Epistemic Boolean Network Game is a tuple $G = \langle M, \mathsf{S}, \gamma, g\rangle$, where $M = \langle A, R\rangle$ is a network, S is a collection of *available strategies*, $\gamma : A \to \mathcal{L}$ is a *goal profile* and g is an initial global state. Given a game G we construct an epistemic iterated network $\langle A, R, W, V\rangle$ satisfying the following requirements.

- A and R are as in M.

- W is the set of possible strategy profiles. A strategy profile s is a possible strategy profile if $s(a)$ is a strategy for a which is available to a and $s(a) \in \mathsf{S}$ for every $a \in A$.
- V is the valuation defined such that for every $s \in W$, s satisfies s.

Our epistemic states are all the possible games that could be played, given S.

Since a game G fully determines an epistemic iterated network, we write $G, s, a \vDash \varphi$ for $M, V, s, a \vDash \varphi$. We then define the *utility* of a strategy profile for a as

$$u_a(s) = \begin{cases} 1 & \text{if } G, s, a \vDash \gamma(a) \\ 0 & \text{otherwise.} \end{cases}$$

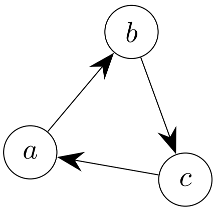
The game as defined assumes the network structure, every players' initial state and the available strategies are all common knowledge. We can modify the construction of the epistemic iterated network to avoid the latter two. Preventing player's knowing the network structure would require more work, but might be possible by letting W consist of combinations of networks and strategy profiles.

Example 3. *Consider the now familiar three player game depicted in Figure 7.1, supposing that* $\mathit{PROP} = \{p\}$. *Each player has the goal of, from some time in the future, being able to predict their neighbour's move one step in advance. Let* $\mathsf{S} = \{\mathit{TiFT}, \mathit{TaFT}\}$.[1] *Recall that these are the strategies of changing to the opposite of a neighbour's colour, and changing to a neighbour's colour, respectively. Every player has the choice of two strategies. Suppose that* g *is the initial state mapping every player to* p.

We can define the epistemic boolean network game $G = \langle M, \mathsf{S}, \gamma, g\rangle$. *In this game, players must choose either TiFT or TaFT, with the goal of being able to predict their neighbour's next move. Given there are 3 players and each player has two choices for strategy, there are* $2^3 = 8$ *possible strategy profiles. Hence in our constructed epistemic iterated network,* $|W| = 8$. *Figure 7.2a compares the runs of the 8 possible strategy profiles.*

The runs have been shaded to indicate the knowledge of a *with regards to the current strategy profile at each timestep, assuming that the actual current strategy profile is 3.*

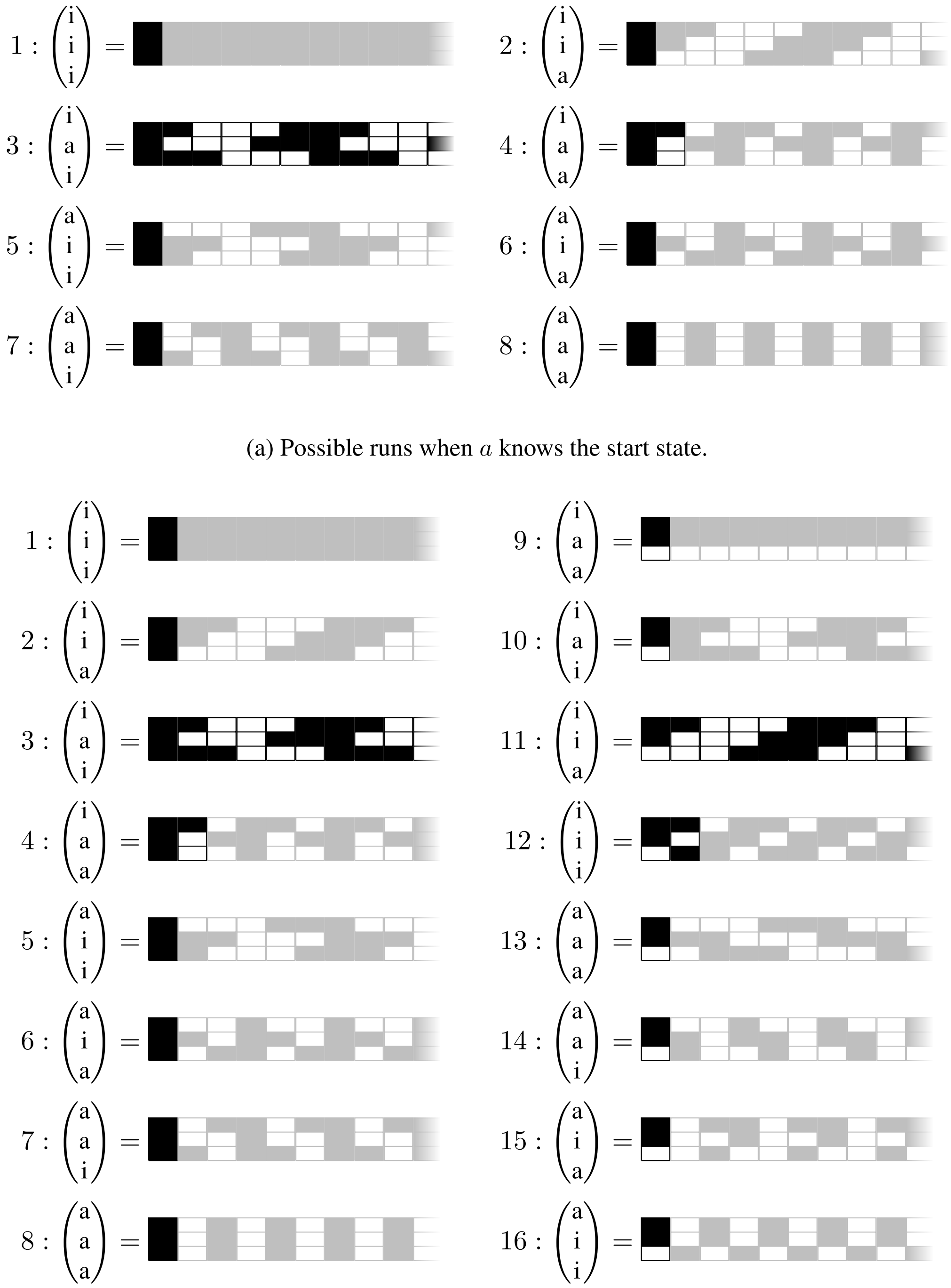
At timestep 0, all the strategy profiles look identical, so a *cannot distinguish them. At timestep 1,* a *sees that* a *has* p *(so rules out 5, 6, 7, 8) and* b *does not (so rules out 1, 2). At timestep 3,* a *sees that* b *has* $\neg p$, *ruling out 4. Hence from timestep 3,* a *knows which strategy profile is in play.*



$$\begin{array}{lll} a: & FG(KX\Diamond p \vee KX\Diamond \neg p) & g(a) = \{p\} \\ b: & FG(KX\Diamond p \vee KX\Diamond \neg p) & g(b) = \{p\} \\ c: & FG(KX\Diamond p \vee KX\Diamond \neg p) & g(c) = \{p\} \end{array}$$

Figure 7.1: Game G_{3KS}.

[1] Note that $|\mathsf{S}|$ is infinite. By $\{\text{TiFT}, \text{TaFT}\}$ we mean S contains all the strategies defined by the general strategies.



(a) Possible runs when a knows the start state.

(b) Possible runs when a does not know the start state.

Figure 7.2: Runs of G_{3KS}. The notation $(i, i, a)^T$ indicates players a, b adopt TiFT and player c adopts TaFT. The actual world is 3. Runs are greyed out as soon as a can rule them out.

By simply waiting, a is able to achieve its goal. In this example, every run but one was eliminated. This is not usual behaviour however. In order to consider an example where multiple runs cannot be eliminated, let us consider a modification in which players do not have total information about the start state.

Example 4. *Suppose players don't know what the start state is. That is, g is not common knowledge. They do, however, know the start state of themselves and their neighbours. Again, consider knowledge from the view of a, and highlight how a can distinguish possibilities when we're in profile 3. Figure 7.2b gives the possible runs.*

Once again we can rule out 5-8 and 13-16 because they do not match a's second move. 1-2 and 11-12 do not match b's second move, so they too can be ruled out. After the third move, 4 and 9 are ruled out since they don't match b's move. However a is never able to distinguish 3 from 10.

In terms of a's goal, this does not matter though! All a needs is to be able to predict b's next move. Whether we're in 3 or 10, after timestep 3 a can predict b's next move since it is the same in both profiles.

In fact this example can be generalised.

Proposition 8. *If $G = \langle M, \mathsf{S}, \gamma, g\rangle$ is an epistemic BNG such that G has finitely many possible strategy profiles*[2] *then for every strategy profile s and player a,*

$$G, s, a \vDash FG(KX\Diamond p \vee KX\Diamond \neg p)$$

and, more specifically, every player will eventually be able to predict the moves of every neighbour.

Proof. Since there are finitely many possible strategy profiles, W is finite. By the Looping Lemma (Lemma 1), each of these strategy profiles will loop after a finite number of steps. Hence, by waiting long enough, any player will know that they are in a loop, and which loop it is. □

Thus for finitely restricted strategy profiles, players can achieve knowledge of their neighbours. The case with infinite possible strategy profiles is more interesting. Indeed, it is possible to construct a set S such that no player ever has knowledge of their neighbour's next moves (take S to be the set of myopic strategies outputting $\neg p$ after n rounds of p, for every $n \in \mathbb{N}$). It would be interesting to pursue which knowledge goals are achievable under these conditions.

7.2.1 Modifications and Refinements

There are a number of ways we could modify this definition of EBNGs to be more flexible in representations of different games. Note that the set S could be chosen as the set of all strategy profiles. In this case, we might abbreviate to $G = \langle M, \gamma, g\rangle$. Of course the most obvious modification is in how W is constructed.

We have already considered preventing players from knowing the start states of every other player, as in Example 4. This seems a reasonable restriction, since players cannot be expected

[2] S being finite is sufficient, but not necessary - we could have infinitely many strategies in S but only finitely many available to each player, like in our example.

to know what non-visible players are doing. Another possibility would be to encode the structure of the network into states of W. In this way, players would be unable to be certain which strategies are available to others, increasing the number of plausible strategy profiles. Inhibiting players knowledge of the network structure mirrors a number of the trials carried out in [14], where players were shown only their local neighbourhood.

One obvious shortcoming of our model is that players do not start with knowledge of their own strategies. This can be seen in Example 3, where a does not rule out runs 5 to 8, even though these have a using a different strategy. It would be easy to fix this by adding the condition that if $s \sim_a^i t$ then $s(a) = t(a)$.

Another inadequacy is the limited nature of higher level knowledge constructions. Standard epistemic models (for example Dynamic Epistemic Logic [20]) allow for higher order knowledge by relying on differences in the accessibility relation between otherwise identical worlds. Again, modifying the construction of W seems a good approach.

No matter how correct a mathematical theorem may appear to be, one ought never to be satisfied that there was not something imperfect about it until it gives the impression of also being beautiful.

GEORGE BOOLE

Chapter 8

Conclusion

Boolean Network Games are powerfully expressive models, useful for modelling a range of situations. We have given some background to the model by defining Boolean Games and Iterated Boolean Games, upon which BNGs are based. We then introduced the BNG model in detail, discussed a number of preliminary properties and examples and connected BNGs to IBGs through an effective translation between each model. Extensions to BNGs, such as the addition of secret states and an epistemic dimension, were also considered. We have raised a large number of questions, many of which remain unanswered. Many of these provide interesting paths for future research.

For example, the difficulty of correctly constructing a conformity-like game, while ruling out trivial solutions, would be interesting to pursue. Our proposals all suffer from some level of triviality. For example, the introduction of a random player still allows for trivial solutions which take into account that player's existence (the ANTIROGUE strategy). A potential avenue to consider is restrictions on players' strategies, to prevent them from being trivial. Capturing the essence of a trivial machine strategy seems difficult however.

Finding a general solution to the diversity game (in networks where this is possible) would also be a fascinating task. As we noted, no general strategy exists which ensures diversity on arbitrary bipartite networks. Our counter-example showing this, C_{2n}, also shows that not every network has a general strategy ensuring diversity. It would be interesting to isolate those networks for which a general strategy exists, and also consider which networks a given general strategy ensures diversity upon.

While the translation of BNGs to IBGs grounds BNGs into the existing literature well, there is some room for further work. Due to the exponential increase in formula size for the BNG to IBG translation, it does not run in polynomial time. As such, a number of complexity results from the IBG literature are not able to be immediately applied. Determining the complexity of problems such as Nash equilibrium existence would be a good further task. One fundamental flaw in our work is that we have not established if existence of Nash equilibria is conserved under our translations. While outcomes of strategies are preserved, it is possible that BNGs with no Nash equilibria are translated to IBGs with Nash equilibria (and in fact in the reverse case this seems probable). Determining whether or not this is the case, and refining the translations to avoid the problem, would be good to consider.

It would also be interesting to investigate the logic of Boolean Network Games. A number of the properties of BNG models (such as looping valuations) seem unlikely to be modally definable, but it would be good to have formal arguments to this effect.

Refining our constructions of modal strategies, and incorporating secret states more carefully into the BNG model seems likely to be a fruitful path to pursue. While it seems unlikely that games in which players have different strategies could be captured by a set of valid formulas (as was the case with IBGs), it nonetheless would be good to establish this.

Refining the EBNG model and investigating its impact on the playing of BNGs presents an exciting direction of research. Addding an epistemic element allows for much greater expressivity on the part of the games. Investigating how EBNGs should be implemented, and comparing different models of epistemic states will be a good area to consider.

We have presented a survey of BNGs and shown the model to be expressive. There is clearly much more to do, however, in this fertile new construction.

Bibliography

[1] J. Richard Bchi. Weak Second-Order Arithmetic and Finite Automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

[2] J. Richard Bchi. On a decision method in restricted second order arithmetic. In *The Collected Works of J. Richard Bchi*, pages 425–435. Springer, 1990.

[3] Kenneth G. Binmore and Larry Samuelson. Evolutionary stability in repeated games played by finite automata. *Journal of economic theory*, 57(2):278–305, 1992.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, August 2002.

[5] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jrme Lang, and Bruno Zanuttini. Boolean games revisited. In *ECAI*, volume 141, pages 265–269, 2006.

[6] Pradeep Dubey and Martin Shubik. A theory of money and financial institutions. 28. The non-cooperative equilibria of a closed trading economy with market supply and bidding strategies. *Journal of Economic Theory*, 17(1):1–20, 1978.

[7] Paul E. Dunne and Wiebe van der Hoek. Representation and complexity in boolean games. In *Logics in Artificial Intelligence*, pages 347–359. Springer, 2004.

[8] Thomas gotnes, Paul Harrenstein, Wiebe van der Hoek, and Michael Wooldridge. Boolean Games with Epistemic Goals. In Davide Grossi, Olivier Roy, and Huaxin Huang, editors, *Logic, Rationality, and Interaction*, number 8196 in Lecture Notes in Computer Science, pages 1–14. Springer Berlin Heidelberg, 2013.

[9] Julian Gutierrez, Paul Harrenstein, and Michael Wooldridge. Iterated boolean games. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 932–938. AAAI Press, 2013.

[10] Julian Gutierrez, Paul Harrenstein, and Michael Wooldridge. Iterated boolean games. *Information and Computation*, 2015.

[11] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean games. In *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge*, pages 287–298. Morgan Kaufmann Publishers Inc., 2001.

[12] Ian Hodkinson and Mark Reynolds. 11 Temporal logic. In Johan Van Benthem and Frank Wolter Patrick Blackburn, editor, *Studies in Logic and Practical Reasoning*, volume 3 of *Handbook of Modal Logic*, pages 655–720. Elsevier, 2007.

[13] Johan Anthony Wilem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.

[14] Michael Kearns, Siddharth Suri, and Nick Montfort. An Experimental Study of the Coloring Problem on Human Subject Networks. *Science*, 313(5788):824–827, August 2006.

[15] Alexander Rabinovich. A proof of kamps theorem. 2014.

[16] Jeremy Seligman and Declan Thompson. Boolean Network Games and Iterated Boolean Games. In *Logic, Rationality, and Interaction*, pages 353–365. Springer, 2015.

[17] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012.

[18] Wolfgang Thomas. Star-free regular sets of -sequences. *Information and Control*, 42(2):148–156, August 1979.

[19] Wiebe Van Der Hoek, Nicolas Troquard, and Michael Wooldridge. Knowledge and control. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 719–726. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[20] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Pieter Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

Appendix A

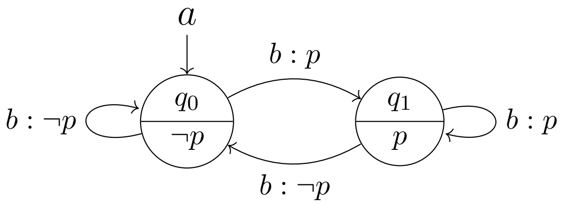
List of BNG Strategies

A.1 Single Variable Strategies

A number of these strategies are named for strategies in [3]. These names may be somewhat misleading, since for our purposes a strategy is independent of a player's goal, whereas in [3], the outputs of strategies are C (cooperate) or D (defect), and so are more tied to a player's goal.

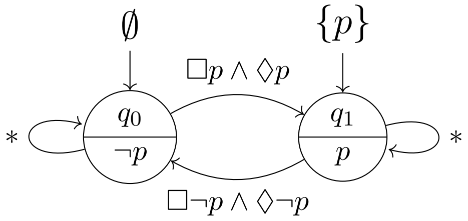
A.1.1 TIT-FOR-TAT (TiFT)

- Single neighbour variant.
- Named for the strategy in [3] that it's similar to.
- Player starts in q_0.
- Player copies their neighbour's move.



A.1.2 Generalised TIT-FOR-TAT (TiFT$_G$)

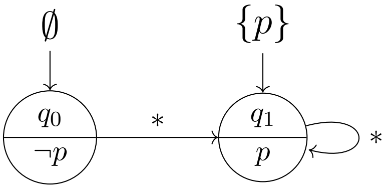
- Multi-neighbour variant.
- Named for the strategy in [3] that it's similar to.
- Player starts in q_0 or q_1 dependent on their start state.
- If player have at least one neighbour, and every neighbour agrees, then player copies all neighbours.



$\emptyset$
$\{p\}$
$\Box p \wedge \Diamond p$
q_0
$\neg p$
q_1
p
$*$
$*$
$\Box \neg p \wedge \Diamond \neg p$

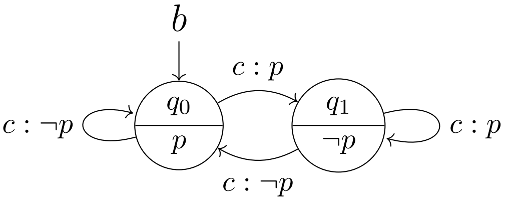
A.1.3 ALL-P

- Multi-neighbour variant.
- Player starts in q_0 or q_1 dependent on their start state.
- Player changes to p as soon as possible, and stays there.



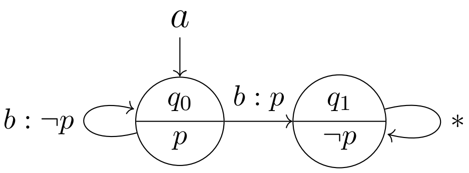
A.1.4 TAT-FOR-TIT (TaFT)

- Single neighbour variant.
- Player starts in q_0.
- Named for the strategy in [3], even though this is slightly different. Really, this is named for TIT-FOR-TAT.
- Player changes to the opposite of its neighbour.



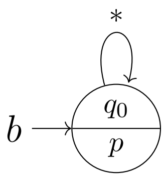
A.1.5 TWEETYPIE

- Single neighbour variant.
- Player starts in q_0.
- Player stays as p until their neighbour plays p, then player endlessly chooses $\neg p$.



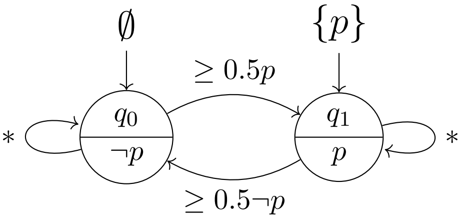
A.1.6 ALLP (variant)

- Multi-neighbour variant.
- Not general.
- Player starts in q_0 and stays there outputting p indefinitely.

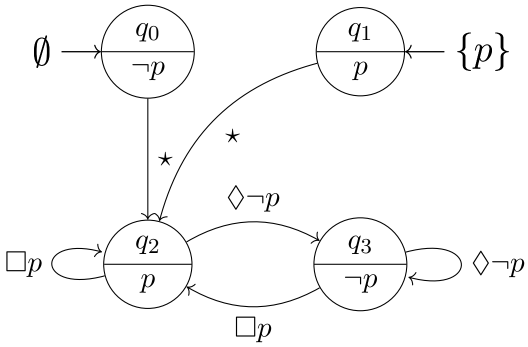


A.1.7 Majority TIT-FOR-TAT (TiFT$_M$)

- Multi-neighbour variant.
- Player starts in q_0 or q_1 dependent on their start state.
- Player copies neighbours if at least have of them have the same state.

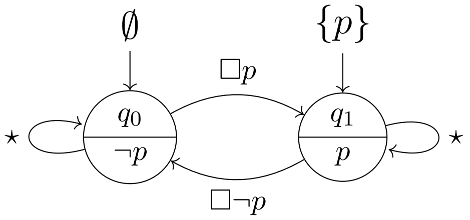


A.1.8 ANTIROGUE



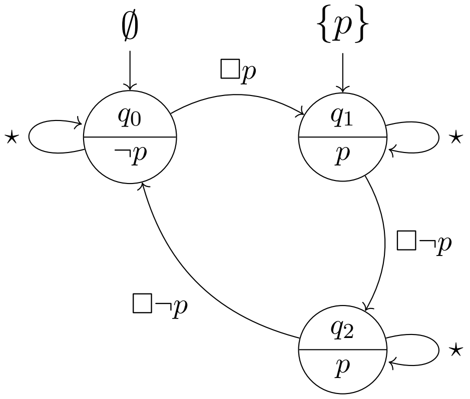
- Multiple player variant.
- Player starts in q_0 or q_1 dependent on their start state.
- Player changes to p as soon as possible, and stays there unless a neighbour has $\neg p$.

A.1.9 Unstable TIT-FOR-TAT (TiFT$_U$)



- Multiple player variant.
- Player starts in q_0 or q_1 dependent on their start state.
- Player copies what all neighbours play, or stays the same. Differs from TiFT$_G$ in that if a player has no neighbours it will alternate indefinitely.

A.1.10 (Unstable) TIT-FOR-FOR-TAT (TiFFT)



- Multiple player variant.
- Player starts in q_0 or q_1 dependent on their start state.
- Like TiFT$_U$ except player gives two “chances” for $\neg p$ before changing.

List of Figures