Algebraic Methods in Game Theory

by

Ruchira Sreemati Datta

B.S. (California Institute of Technology) 1991
M.S. (University of California at Berkeley) 2002

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Mathematics

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Bernd Sturmfels, Chair
Professor Alberto Grunbaum
Professor Laurent El Ghaoui

Fall 2003

Algebraic Methods in Game Theory

Abstract

Algebraic Methods in Game Theory

by

Ruchira Sreemati Datta
Doctor of Philosophy in Mathematics

University of California at Berkeley

Professor Bernd Sturmfels, Chair

In this dissertation we apply algebraic methods to game theory. The central objects of study in game theory, Nash equilibria, can be characterized in several ways. We focus on their characterization as the solutions to certain systems of polynomial equations and inequalities. Thus we bring to bear the techniques of commutative algebra, algebraic geometry, and combinatorics used in solving polynomial systems. In Chapter 1 we give a brief overview of the game theory we need. We restrict attention to games with a finite number of players each with a finite number of pure strategies. We mostly consider noncooperative normal form games, although we discuss finite-horizon extensive form games briefly and also mention cooperative games.

In Chapter 2 we prove the universality of Nash equilibria. Every real algebraic variety is isomorphic to the set of totally mixed Nash equilibria of some game with 3 players, and also of some game with N players in which each player has two pure strategies. Our proof is constructive. The numbers of pure strategies in the game with 3 players, and the number of players in the game with two pure strategies each, are polynomial in the degrees of the equations. Thus the problem of computing Nash equilibria in general is equivalent to the problem of finding the real roots of a system of polynomial equations.

In Chapter 3 we prove a theorem computing the number of solutions to a system of equations which is generic subject to the sparsity conditions embodied in a graph. We apply this theorem to games obeying graphical models and to extensive-form

games. We define *emergent-node tree structures* as additional structures which normal form games may have. We apply our theorem to games having such structures. We briefly discuss how emergent node tree structures relate to cooperative games.

In Chapter 4 we discuss how to compute all Nash equilibria of a game using computer algebra. We find that polyhedral homotopy continuation is the most efficient available method in practice. It also has the advantage of being naturally parallelizable. We discuss further directions for developing algebraic algorithms for computing Nash equilibria.

Professor Bernd Sturmfels
Dissertation Committee Chair

Dedication

To Śrīla Bhakti Pramod Purī Mahārāj,

by whose grace it was possible.

namo gurave namaḥ

Contents

List of Figures

Acknowledgements

As I'm about to file my dissertation, I'm bursting with more gratitude than an Oscar-winning actress, and in the same way I have many more people to thank than I possibly can. In the interests of brevity, I can only mention by name the people who helped me most directly with this dissertation, but there have been many others along the way whose friendly support I appreciate. I thank them all.

I thank INFORMS, the Institute for Operations Research and the Management Sciences, for their permission to present material similar to [Dat03a] (for which their copyright is hereby noted) as Chapter 2 of this dissertation, and the ACM, the Association for Computing Machinery, for their permission to present much of [Dat03b] (for which their copyright is hereby noted) as Chapter 4 of this dissertation.

Firstly I must thank Professor Michael Aschbacher, who introduced me to the beautiful world of abstract algebra in his crystal-clear lectures and who was my undergraduate adviser at Caltech. Towards the end of my senior year, he met with me in his office and encouraged me to pursue a career in mathematics. His words have helped sustain me through my doctoral studies, and I am sincerely grateful.

Next I must sincerely thank Professor Robion Kirby who, as Graduate Vice Chair, encouraged me to come to Berkeley. (I also remember his differential topology class with great fondness!) Many people in the Berkeley mathematics department have helped create a friendlier environment. In particular, among my fellow graduate students I would like to thank Alex Gottlieb for being such a great officemate and Benson Farb for making exam grading go faster; among the faculty I would like to thank Professor George Bergman for frequently asking how I was doing; and among the staff I would like to thank Dave Hernes, the building manager, and Janet Yonan, Marsha Snow, Catalina Cordoba and Thomas Brown in the graduate office for their great service. Really there are too many people to list; I wish I could thank them all.

I would like to acknowledge support by National Need Fellowships from the U.S. Department of Education, and I would like to thank Dr. Robert Sonderegger and SRC Systems Inc. for their financial support, during portions of my doctoral studies.

I was a Graduate Student Instructor for two years, and I would like to thank the professors, my fellow GSIs, and the students in the classes I taught, from all of whom I learned a lot.

I would especially like to thank Professor Ken Ribet. As Graduate Vice Chair, his encouragement was extremely helpful at a crucial juncture, and I am deeply grateful. I would also like to thank Professor David Eisenbud for the wonderful course he taught, along with Greg Smith, from his book on commutative algebra.

I would like to thank Professor Pravin Varaiya in the Electrical Engineering department as well as Duke Lee and the rest of the gang with whom I worked on the Wireless Token Ring Project. We had a great time! I also would like to thank Professor Raja Sengupta in the Civil & Transportation Engineering department and the rest of the Systems Theoretical & Applied Research group, as well as the Mixed Initiative Control for Automata project. I really enjoyed learning about and working in systems theory, and I hope to return to the subject in the very near future. I received support from both these projects during my doctoral studies.

I would like to thank Professor Laurent El Ghaoui for his great class on convex optimization (and especially for inviting me to take one class period to give a presentation on polynomial optimization), for many friendly discussions of interesting problems, and especially for being on my dissertation committee (about which more below).

I would like to thank Professor Mark Haiman. He discussed beautiful mathematical problems with me and encouraged me at a critical time. He was also on my qualifying exam committee.

I would like to thank Professor Alberto Grunbaum. He has known me throughout my doctoral studies, from my first year when I took his differential equations class, and his sincere good wishes have meant a lot to me. He has also supported me financially. It was partially through his support that I purchased the laptop, described in Chapter 4, which I used in researching and writing this dissertation. He also supported me during the research that led to Chapter 2 and my paper [Dat03a]. Finally I thank him for being on my dissertation committee.

I would like to thank Professor Reinhard Laubenbacher of the Virginia Bioinformatics Institute at Virginia Tech. We've had many interesting discussions about computational biology. He also supported me during the research that led to Chapter 4 and my paper [Dat03b]. I would also like to thank Professor Richard Fateman in the Computer Science Department here at Berkeley, my advisor for my master's degree in computer science. It was he who suggested I submit the paper [Dat03b] to ISSAC 2003, and he supported my attendance there to present it.

I would like to thank Professor Lior Pachter. I enjoyed the research seminar on computational biology he ran with Professor Dick Karp. We've had many interesting discussions and he's been a great friend. I'm also learning a lot in the seminar on the Mathematics of Phylogenetic Trees he's running now with Bernd. Lior supported me during the research that led to Chapter 3.

I would like to thank my mathematical siblings. Of those who came before me, I know Rekha Thomas, John Dalbec, Jesús De Loera, Serkan Hoşten, Diane Maclagan, Ezra Miller, Laura Matusevich, Amit Khetan, and Mike Develin. I've interacted with them to varying degrees, and their friendly advice, encouragement, and help in many substantial ways has meant a lot to me. I especially remember a conversation with Ezra during the banquet at the AMS meeting, Mathematical Challenges of the 21st Century, at UCLA in 2000 (before he became my mathematical sibling!), which had a critical influence on my subsequent path. Serkan suggested I submit my paper [Dat03a] to Mathematics of Operations Research, and Jesús suggested the lectureship at UC Davis where I'm going next. I would also like to thank my younger mathematical siblings, from whom I've learned a lot: my officemate Seth Sullivant, Chris Hillar, Dave Speyer, Nick Eriksson, and Josephine Yu, as well as my mathematical "nieces" Maya Ahmed and Ruriko Yoshida. I'm privileged to be in their company, and I'm sure there are many more to come!

At last I would like to thank my thesis advisor, Professor Bernd Sturmfels. He took a chance on me and I will always strive to justify his faith. Bernd has helped me in more ways than it ever even occurred to me to expect. He has completely changed my idea (and set the gold standard) of what a thesis advisor should be, and I think I appreciate him all the more deeply for that reason. How can I possibly thank Bernd enough?

Bernd first agreed to let me work with him in the fall of 2001. Shortly afterward,

at a post-seminar dinner, he regaled us with the tale of his recent visit to New York. His friend, Professor Dave Bayer of Columbia, was then acting as mathematical advisor during the making of the film *A Beautiful Mind*. Dave had taken Bernd to visit the set, where he had met Russell Crowe and Ron Howard. I was duly awed. (Thus Dave was responsible for introducing Bernd, and indirectly me, to game theory, for which I must thank him!)

One Saturday morning in late March of 2002, Bernd emailed me very excitedly. Could I go to campus and pick up the copy of John Nash's 1951 paper "Non-cooperative games" he had put in my cubbyhole? Bernd was in the midst of preparing a series of lectures on solving systems of polynomial equations, in which he wanted to include game theory, and he asked me to write up Nash's somewhat terse example of the Three-Man Poker Game. As Bernd said, "First the book, then the movie, now the Gröbner bases!" This is just one example of the energy and enthusiasm Bernd brings to all of his myriad mathematical interests, and conveys to those around him.

Shortly afterwards we met with Professor El Ghaoui and his student Aswin Ganesan, who were also studying game theory. Professor Emeritus David Gale graciously agreed to meet with all of us, and we had a very interesting discussion, for which I thank him.

At this time Bernd suggested to me the problem of proving the universality theorem, the main subject of Chapter 2 and [Dat03a]. Bernd presented his lectures on polynomial equations in a course (one of several great courses I took from him), and later that spring at Texas A&M (as part of the Conference Board Mathematical Series), where he took me along. That summer Bernd gave me the privilege of helping him with the resulting book [Stu02]. (This partially funded the purchase of the abovementioned laptop.)

I showed an early draft of Chapter 2 to Professor Andrew Maclennan in the economics department at the University of Minnesota. His enthusiastic reception was extremely encouraging. He suggested that I apply Nash's beautiful theorem of [Nas52]. During and after my visit with him that September, we had very valuable and extensive discussions on the presentation of [Dat03a]. I thank him for his generous help.

Bernd also suggested that I experiment with computing Nash equilibria, which led to Chapter 4. I would like to express my gratitude to Andrew McLennan and Ted Turocy, Mike Stillman, Gert-Martin Greuel, and Jan Verschelde for generously taking the time to personally discuss with me the use of their software packages Gambit [MMT], Macaulay2 [GS], Singular [GPS01], and PHC [Ver99] respectively.

Bernd also suggested that I prove a Bernstein bound on the number of totally mixed Nash equilibria of graphical games, which led to Chapter 3. During the fall of 2003 I have also sometimes attended Professor Shachar Kariv's course on noncooperative game theory in the economics department, which led me to include the treatment of extensive-form games in Chapter 3. Shachar also very generously agreed to take time out of his schedule to read the draft of my dissertation, for which I sincerely thank him.

Finally I must thank my family: my mother Debbie Datta, my father Robin Datta, and my sister Saheli S. R. Datta, for their unwavering support, morally, financially, physically, spiritually, and in every way through all these years. They have never doubted that I had a contribution to make as a mathematician. Without their faith and love I could never have completed my studies.

Most of all I must thank my Gurudev, my spiritual teacher, Śrīla Bhakti Pramod Purī Mahārāj. I feel his grace upon me every day of my life, which makes possible everything that I achieve.

Colophon. This dissertation was written almost entirely with free software (the major exception being the commercial fonts). I have to thank the many people responsible for Debian Gnu/Linux (including TEX tools); Bram Moolenaar for Vim; Donald Knuth for TEX (I'm a big fan of the TeXbook!) and Leslie Lamport for LATEX; and the former Berkeley graduate students who created the `ucthesis` style, for making typesetting this dissertation as painless as possible. I especially must thank Alan Hoenig for his beautiful book *TEX Unbound* and the `vfinst` and `mathinst` font utilities. I used `mathinst` to make a mathematical font family of Monotype Bembo Semibold (from Agfa-Monotype), MathTime (from Y&Y), Chantilly (from Softmaker, similar to Gill Sans), Typewriter (from the Electronic Font Foundry), and a few others, with which I typeset this dissertation.

Chapter 1

Introduction

In 1654, a gambler, the Chevalier de Méré, posed two questions on games. In the ensuing exchange of letters between Blaise Pascal and Pierre Fermat, probability theory—or as Pascal more evocatively called it, *la géometrie du hasard* ("the geometry of chance")—was born [Pas63]. Returning to the roots of probability theory, Émile Borel turned his attention to games, and between 1921 and 1927 wrote a seminal series of papers in which he introduced the abstract formalism which we now call game theory [Bor21], [Bor24], [Bor27].

Decision theory, a special case of game theory, was used by Abraham Wald to create a firm foundation for mathematical statistics [Wal50], as described by Blackwell and Girshick [BG54]. Game theory also provides a unifying framework for approaching artificial intelligence [Wel95]. Since John von Neumann and Oskar Morgenstern published their influential *Theory of Games and Economic Behavior*, game theory has taken on an increasingly central role in mathematical economics; indeed in his textbook [Kre90] David Kreps describes microeconomics through the lens of game theory. Maynard Smith applied game theory to evolutionary biology to create *evolutionary game theory* [Smi68]. Thus, game theory is a powerful abstraction which illuminates many fields.

So what is game theory? We summarize the concepts we need below, but first we will very briefly go over a few concepts from algebra and combinatorics that we will need.

1.1 Solving Polynomial Systems

A *monomial* in n variables $x_1, \ldots, x_n$ is an expression of the form $x_1^{d_1} \ldots x_n^{d_n}$ for some nonnegative integers $d_1, \ldots, d_n$. It is *squarefree* if $d_i \leq 1$ for all i. The *degree in x_i* of this monomial is d_i, and its *total degree* is $\sum_{i=1}^{n} d_i$. The product of two monomials $x_1^{d_1} \ldots x_n^{d_n}$ and $x_1^{e_1} \ldots x_n^{e_n}$ is $x_1^{d_1+e_1} \ldots x_n^{d_n+e_n}$.

A *polynomial* in n variables $x_1, \ldots, x_n$ with coefficients in a field K is a finite sum of *terms*. Each term is of the form cm for some $c \in K$ and some monomial m in $x_1, \ldots, x_n$. The set of polynomials in n variables is a vector space whose basis is the set of monomials in n variables. Since we can also multiply monomials together, we can extend this to define a product on the space of polynomials, using commutativity, associativity, and distributivity. This makes the set of polynomials in n variables with coefficients in K into a *commutative ring*, and the study of such objects is the subject of *commutative algebra*.

We can *evaluate* a polynomial $f = \sum c_{d_1 \ldots d_n} x_1^{d_1} \cdots x_n^{d_n}$ at a point $(a_1, \ldots, a_n) \in K^n$ by *substituting* $a_1, \ldots, a_n$ for the variables $x_1, \ldots, x_n$ respectively, obtaining an expression $\sum c_{d_1 \ldots d_n} a_1^{d_1} \cdots a_n^{d_n}$ and carrying out all the multiplications and additions in the field K. We denote the resulting element of K as $f(a_1, \ldots, a_n)$.

A *polynomial equation* is an expression $f = g$ for some polynomials f and g. A point $x \in K^n$ satisfies this equation if $f(x) = g(x)$. Since this is equivalent to $f(x) - g(x) = 0$, we can always write a polynomial equation as $p = 0$ for some polynomial p. A point satisfying this equation is called a *root* of p. A polynomial in one variable with coefficients in K need not have any roots in K. But there is always a field containing K, called the *algebraic closure $\bar{K}$ of K*, such that every nonconstant univariate polynomial with coefficients in $\bar{K}$ has a root in $\bar{K}$. The field of complex numbers is the algebraic closure of the field of real numbers.

A *system of polynomial equations in n variables* over K is a finite set of polynomial equations in n variables over K, and a point in K^n satisfies, or is a root of, this system if it satisfies all the constituent equations. The study of solution sets of polynomial equations is called *algebraic geometry*. Recent years have seen a renaissance in *computational algebraic geometry*; the interested reader is referred for example to

[CLO97].

1.2 Concepts From Combinatorics

A *graph* is given by a set of elements, called *nodes* or *vertices*, together with a set of pairs of nodes, called *edges*. We will only consider finite graphs, that is, graphs with a finite set of vertices. If the edges are ordered pairs, we say the graph is *directed*, and each edge goes *from* its first element (or source, or tail) *to* its second element (or target, or head). If the edges are unordered pairs (i.e., sets of two nodes), we say the graph is *undirected*, and each edge goes *between* its elements. We will only consider directed graphs. An edge from a node to itself is called a *self-loop*. The graphs we will consider will all be *simple*, i.e., they will have no self-loops and will not have multiple edges in the same direction between the same pair of nodes. A *path* in a graph is a sequence $v_0 v_1 \ldots v_n$ of nodes such that (v_i, v_{i+1}) is an edge in the graph for each $i = 0, \ldots, n-1$. We say the path goes from v_0 to v_n, or between v_0 and v_n.

A *(rooted) tree* is a special kind of directed graph. It has a distinguished node called the *root*, and its edges are called *branches*. No edge goes to the root. Every other node v has a unique edge going to it. The node w at the tail of this edge is called v's *parent*, and v is said to be w's *child*. For each node v, there is a unique path in the tree from the root to v. We can find this path by going backward and finding v's parent, then the parent of its parent, and so forth, until we reach the node with no parent—the root. A node with no children is called a *leaf*. If there is a (directed) path in the tree from a node v to a node w, we say v is an *ancestor* of w and w is a *descendant* of v. Each node is an ancestor and a descendant of itself, and also a descendant of the root. For each node v in a tree, the *subtree* below v has the descendants of v as vertices and the same edges between these vertices as were in the original tree. The node v itself is the root of this subtree.

A *polytope* is the set of solutions to some system of (nonstrict) linear inequalities in Euclidean space. If this system is irredundant (that is, no inequalities can be dropped without changing the polytope) and there are points in the polytope which satisfy one of the inequalities strictly, then the set of points in the polytope where it holds with equality is called a *facet* of the polytope. One can consider facets of the facets, facets of the facets of the facets, and so forth; all of these are called *faces*. Each such descending chain of faces, where each face is a facet of the previous one, eventually terminates in a face which is a single point, or *vertex*.

Given a finite set of points $v_0, \ldots, v_d$ in a real vector space such that $v_1 - v_0, \ldots, v_d - v_0$ are linearly independent, the *simplex* with those points as vertices is the set

$$\left\{ \sum_{i=0}^{n} \lambda_i v_i \colon 0 < \lambda_i \in \mathbb{R} \text{ for } i = 0, \ldots, n \text{ and } \textstyle\sum_{i=0}^{n} \lambda_i = 1 \right\}.$$

The (d-dimensional) *probability simplex* over a finite set of events $v_0, \ldots, v_d$ is obtained by identifying v_0 with the origin and $v_1, \ldots, v_d$ with the unit coordinate vectors in $\mathbb{R}^d$. It is the polytope $\{(\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^d \colon \lambda_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{d} \lambda_d \leq 1\}$.

The notion of the *determinant* of a matrix M is familiar from linear algebra: it is the sum of certain signed products of entries of M. We will also need the notion of the *permanent* of M. It is the sum of the same products of entries of M, but without the additional sign factors.

1.3 Game Theory Concepts

The concepts we describe in this section can be found in a standard game theory text such as [OR94]. However, in some cases we use simplified notation for the restricted situations we will consider.

Game theory is the study of strategic interaction. Such interaction takes place between multiple agents in a single setting, or *environment*. An *agent* is an entity which can receive *information* about the state of the environment (including itself and other agents), take *actions* which may alter that state, and express *preferences* among the various possible states. These preferences are encoded for each agent by a *utility function*, a mapping from the set of all states to $\mathbb{R}$. Its value for a particular state is the *utility* of that state for the agent. The agent prefers one state to another if its utility is greater, and is indifferent between them if their utilities are equal.[1] Changes

[1] Instead of specifying the utility of each state for each agent, one might specify the *change* in utility,

in the state of the environment may also occur spontaneously (i.e., not due to the actions of any of the agents). A *strategy* is a (possibly stochastic) rule for an agent to choose an action at every point when the agent may act, given the available information. A *rational* agent is one whose strategy maximizes its expected utility under the circumstances.

We will restrict attention to games which take place in a finite number of time steps between a finite number of agents, each of which has a finite number of possible actions. The agents are called *players*, and whenever they take an action they are said to *move*. A spontaneous change in the state of the environment is called a *move by nature*. The game is over when no player (including nature) has any possible actions. The state of the environment at such a terminal stage is called an *outcome*. Generally preferences are specified only over outcomes, not at intermediate stages of the game.

The first type of game we will consider is the *normal-form game*. In a normal-form game, there is only one time step, at which all the players move simultaneously. We denote the set of players by $I = \{1, \ldots, N\}$. The actions a player can take are called *pure strategies*. (Thus a pure strategy is the simplest possible rule: it always chooses one particular action.) We associate to the players finite disjoint sets of pure strategies $S_1, \ldots, S_N$. For each i let $d_i = |S_i| - 1$. We write the set S_i as $\{s_{i0}, \ldots, s_{id_i}\}$. We write $S = \prod_{i \in I} S_i$. Game play consists of the collective choice of an element of S by the players: each player i moves by choosing an element of S_i. We identify S as the set of possible outcomes. We denote by $u_i(s)$ the utility for player i of the outcome $s \in S$. Thus, the game is completely specified by the number N of players, the sets S_i of pure strategies, and the utility functions (or *payoff functions*) $u_i \colon S \mapsto \mathbb{R}$.

A player may move stochastically rather than deterministically. In that case the player is said to execute a *mixed strategy*. The mixed strategy specifies the probability with which the player chooses each possible action. The set Σ_i of mixed strategies

---

or *marginal utility*, which accrues to each agent upon each *transition* between states. Clearly any utility function induces a marginal utility function, but unless one imposes additional conditions a marginal utility function may not induce a utility function. Such a marginal utility function, which one might call *intransitive*, could still be a useful model of reality. For example, one wouldn't necessarily feel the same about being laid off and then immediately rehired as if one had simply continued in the same position. However, we will not consider such intransitive marginal utility functions any further.

of player i is the set of all functions $\sigma_i \colon S_i \mapsto [0,1]$ with $\sum_{s_{ij} \in S_i} \sigma_i(s_{ij}) = 1$. That is, it is the d_i-dimensional probability simplex. We write $\Sigma = \prod_{i \in I} \Sigma_i$. An element σ of Σ, which specifies the strategies executed by all the players, is called a *strategy profile*. If the players execute the strategy profile σ, then the probability of outcome s is $\sigma(s) = \prod_{i=1}^{N} \sigma_i(s_i)$. The *expected* utility for player i of the strategy profile σ is given by multilinearity as $u_i(\sigma) = \sum_{s \in S} u_i(s)\,\sigma(s)$.

When considering how agent i should behave, it will be convenient to separate out i's own strategy, over which i has control, from the strategies of all the other players. We write $\Sigma_{-i} = \prod_{j \in I - \{i\}} \Sigma_j$, and we write σ_{-i} for the image of $\sigma \in \Sigma$ under the projection π_{-i} from Σ onto Σ_{-i}. By abuse of notation, we write $u_i(\tau_i, \sigma_{-i})$ for the ith player's expected payoff from the strategy σ whose ith component is τ_i and whose other components are defined by $\pi_{-i}(\sigma) = \sigma_{-i}$.

We assume *perfect information*: each player knows the complete specification of the game, knows that every player knows, knows that every player knows that every player knows, ad infinitum. That is, the specification of the game is *common knowledge*. Under these circumstances, what is rational behavior? In his landmark paper [Nas50], John Nash answered this question in terms of what is now called *best response*. A best response of player i to the strategy profile σ is a mixed strategy σ_i^* such that $u_i(\sigma_i^*, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i})$ for any other mixed strategy σ_i' of player i. That is, given that all the other players execute the strategy profile σ_{-i}, the mixed strategy σ_i^* maximizes player i's expected utility. A *Nash equilibrium* is a strategy profile which is a best response to itself for all the players. That is, it is a strategy profile σ^* such that for each player i, we have $u_i(\sigma^*) \geq u_i(\sigma_i', \sigma_{-i}^*)$ for every other mixed strategy σ_i' of player i. Nash proved that such an equilibrium always exists.

How can we compute the Nash equilibria of a given game? We need to search the set Σ of strategy profiles, which is a polytope: the product of probability simplices. We can decompose the problem by *stratifying* this polytope: first we look for Nash equilibria in its interior, then in the interiors of its facets, then in the interiors of the facets of those facets, and so forth, until finally we look for Nash equilibria at the vertices of the polytope (that is, pure strategy Nash equilibria). A strategy profile σ lies in the interior of this polytope if $\sigma_i(s_{ij}) > 0$ for every $s_{ij} \in S_i$, for every i. Such a strategy profile is called *totally mixed*. Note that a totally mixed Nash equilibrium *need not* exist.

Suppose we had a procedure to find the totally mixed Nash equilibria, that is, those lying in the interior of the polytope, for any game. Then we could also find the Nash equilibria lying in the interior of any facet of the polytope. Such a facet is defined by an equation $\sigma_i(s_{ij}) = 0$ for some player i and some pure strategy s_{ij} of that player. We apply our procedure to find the totally mixed Nash equilibria of the game induced by restricting to the pure strategy sets $S_1, \ldots, S_i - \{s_{ij}\}, \ldots, S_N$. For each strategy profile σ we obtain in this way, we check whether $u_i(\sigma_i, \sigma_{-i}) \geq u_i(s_{ij}, \sigma_{-i})$; if so, then σ is a Nash equilibrium of the original game. Thus, once we have this procedure we can find all Nash equilibria by a divide-and-conquer approach. Of course, here we have not dealt with efficiency questions, but only the existence of a finite algorithm. Indeed, even when the players only have two pure strategies each, just the number of vertices of the polytope, 2^N, is exponential in the number N of players, let alone the number of faces of all dimensions. Thus the combinatorial explosion keeps this algorithm from being tractable.

So we concentrate our attention on the totally mixed Nash equilibria. We observe that for a totally mixed strategy profile σ to be a Nash equilibrium, it is necessary and sufficient that for each player i we have $u_i(s_{ij}, \sigma_{-i}) = u_i(s_{ik}, \sigma_{-i})$ for any pure strategies $s_{ij}, s_{ik} \in \Sigma_i$. These equations are called the *indifference equations* for player i. The sufficiency is clear. For the necessity, suppose to the contrary that $u_i(s_{ij}, \sigma_{-i}) > u_i(s_{ik}, \sigma_{-i})$. Define σ_i' by

$$\sigma_i'(s_{il}) = \begin{cases} \sigma_i(s_{ij}) + \sigma_i(s_{ik}), & l = j \\ 0, & l = k \\ \sigma_i(s_{il}), & \text{otherwise} \end{cases} .$$

Then since $\sigma_i(s_{ik}) > 0$, we have $u_i(\sigma_i', \sigma_{-i}) = u_i(\sigma) + \sigma_i(s_{ik})\left(u_i(s_{ij}, \sigma_{-i}) - u_i(s_{ik}, \sigma_{-i})\right) > u_i(\sigma)$, a contradiction.

So we have a system of $\sum_{i=1}^N d_i$ equations, $u_i(s_{ij}, \sigma_{-i}) = u_i(s_{i0}, \sigma_{-i})$ for $j = 1, \ldots, d_i$, for $i = 1, \ldots, N$, in $\sum_{i=1}^N d_i$ unknowns $\sigma_i(s_{ij})$ for $j = 1, \ldots, d_i$, for $i =$

$1, \ldots, N$. (Here we have *dehomogenized*, that is, we have eliminated $\sigma_i(s_{i0})$ by substituting $1 - \sum_{j=1}^{d_i} \sigma_i(s_{ij})$). What we are equating are the expressions $u_i(s_{ij}, \sigma_{-i}) = \sum_{s_{-i} \in S_{-i}} u_i(s_{ij}, s_{-i})\, \sigma_1(s_1) \cdots \sigma_{i-1}(s_{i-1})\, \sigma_{i+1}(s_{i+1}) \cdots \sigma_N(s_N)$, which are multilinear polynomials whose coefficients are the real numbers $u_i(s)$. The (possibly complex) roots of this system are called *quasiequilibria*, and those roots which are totally mixed strategy profiles (that is, which are real with $\sigma_i(s_{ij}) > 0$ and $\sum_{j=1}^{d_i} \sigma_i(s_{ij}) < 1$) are the totally mixed Nash equilibria. The study of systems of polynomial equations is the field of *algebraic geometry*, and the subject of this dissertation is algebraic methods in game theory.

If s_{ij} and s_{ik} are pure strategies of player i, then s_{ij} *strictly dominates* s_{ik} if for all $s_{-i} \in S_{-i}$, we have $u_i(s_{ij}, s_{-i}) > u_i(s_{ik}, s_{-i})$. (Then of course no totally mixed Nash equilibria can exist in the game as specified.) In this case player i, being rational, should never play s_{ik}, so we might as well eliminate it from S_i. After we do this some more pure strategies of some other players may become strictly dominated, and so we iterate. We will always assume that we have already carried out *iterated elimination of strictly dominated strategies* on any game under consideration.

We will also briefly consider *finite horizon extensive-form games*. (See for example [OR94], Chapter 6.) Such a game takes place in a finite number of time steps, at each of which only a single player (possibly nature) may move. (Which player moves, and what actions the player is allowed to take, may depend on what moves were made previously.) Such a game is completely specified as follows. We specify a set of players $I = \{1, \ldots, N\}$, and we specify a *game tree* T: a finitely branching tree of finite depth in which each non-leaf node is labelled by a number in $0, \ldots, N$, each leaf is labelled by an N-tuple of real numbers, and each branch emanating from a (non-leaf) node labelled 0 is assigned a positive real weight, so that the total weight emanating from such a node is 1. (We consider the branches of this tree to be directed away from the root.)

Game play proceeds as follows. Each node of the tree represents a state of the environment. At each time step, if we are at a non-leaf node labelled by i in $1, \ldots, N$, then player i acts by choosing one of the branches emanating from that node. Then the environment undergoes the transition to the node at the end of that branch, and we advance to the next time step. If we are at a non-leaf node labelled by 0, then the environment instead makes a random transition along one of the branches emanating from that node; the probability of each branch is given by its weight. If we are at a leaf node λ labelled by $(u_1(\lambda), \dots, u_N(\lambda))$, then the game is over, and each player i accrues utility $u_i(\lambda)$. Thus, the leaf nodes are the outcomes of the game.

Unless otherwise specified, we will assume perfect information. Not only do all players have common knowledge of the specification of the game, but whenever a player is about to move, that player knows what moves have been made by all the other players (including nature) up to that point.

Every extensive-form game is equivalent to a normal-form game. For each node ν of the game tree, we write $E(\nu)$ for the set of edges emanating from ν. Then the set of pure strategies of player i is

$$S_i = \prod_{\substack{\nu \in T \\ \text{label}(\nu)=i}} E(\nu).$$

Given a pure strategy profile $s \in S = \prod_{i \in I} S_i$, we can compute the probability of each leaf node λ of the game tree. A unique path $\nu_0 \nu_1 \dots \nu_m = \lambda$ leads from the root ν_0 of T to λ. Then $\Pr[\lambda|s] = \prod_{j=0}^{m-1} \Pr[\nu_j \to \nu_{j+1}|s]$, where

$$\Pr[\nu_j \to \nu_{j+1}|s] = \begin{cases} 1, & \nu_j \text{ is labelled by } i \in I \text{ and } s_{i\nu_j} = (\nu_j \to \nu_{j+1}) \\ 0, & \nu_j \text{ is labelled by } i \in I \text{ and } s_{i\nu_j} \neq (\nu_j \to \nu_{j+1}) \\ \text{wt}(\nu_j \to \nu_{j+1}), & \nu_j \text{ is labelled by } 0 \end{cases}$$

and so the utility functions of the normal-form game are given by

$$u_i(s) = \sum_{\substack{\lambda \in T \\ \lambda \text{leaf}}} u_i(\lambda) \Pr[\lambda|s].$$

We note that the game specification implies certain equalities among the numbers $u_i(s)$. If we consider the set of normal-form games with a fixed set of players I and outcomes S to be a linear space with basis $\{u_i(s) \colon s \in S\}$, then the extensive-form games with the same set of players I and a fixed game tree having S as the set of outcomes lie in a linear subspace of this space, given by these equalities. Let A be the set of non-leaf nodes of the tree which are not labelled by 0. Then we can identify S with $\prod_{\nu \in A} E(\nu)$. For any $s \in S$ and $\nu \in A$, we write $s_\nu = s_{i_\nu}$, where i is the label of ν. Suppose $\nu \in A$ is an ancestor of $\mu \in A$. Then ν has a unique child α that is also an ancestor of μ (possibly μ itself). Let β be any other child of ν. If $s, s' \in A$ with $s_\nu = (\nu \to \beta)$ and

$$s'_\nu = \begin{cases} e, & \nu = \mu \\ s_\nu, & \text{otherwise} \end{cases}$$

for some edge $e \in E(\mu)$, then $u_i(s) = u_i(s')$. This is because $\Pr[\lambda|s] = \Pr[\lambda|s'] = 0$ unless λ is a descendant of β or λ is not a descendant of ν, and in either case λ cannot be a descendant of μ. In short, the node μ is never reached, so it doesn't matter which action is chosen there.

If different players act at ν and μ, then there is no way to eliminate this redundancy, but when the same player i acts at ν and μ, we can do so. In this case we replace all the pure strategies which are forced to be equal by a single pure strategy, called a *reduced pure strategy*. See for example [OR94], p. 94.

We note that after iterated elimination of strictly dominated pure strategies, for any node all of whose children are leaves, the payoffs to the player who acts at that node must be equal at all these child leaves. If nature acts at such a node ν whose children are leaves $\lambda_1, \ldots, \lambda_k$, then we can replace ν by a leaf with utilities $u_i(\nu) = \sum_{l=1}^{k} \mathrm{wt}(\nu \to \lambda_l) u_i(\lambda)$ for each $i \in I$. So we assume nature never acts at such nodes.

For extensive-form games, the equilibrium concept can be refined. Each subtree of the game tree induces a new extensive-form game, called a *subgame*. Each pure strategy of the original game induces a pure strategy of each subgame by restriction to that subtree, and thus each strategy profile of the original game induces a strategy profile of each subgame. A strategy profile is a *subgame perfect Nash equilibrium* of an extensive-form game if it induces a Nash equilibrium of each subgame.

We can find a subgame perfect pure strategy Nash equilibrium by *backwards induction*. We construct the pure strategy profile as follows. We perform iterated elimination of strictly dominated strategies. Then at each node all of whose children are leaves, we choose one leaf (recall that the payoffs of all leaves for the player who acts at that node will be the same). We assign this branch to the corresponding component of the pure strategy profile, replace this node by this leaf, and repeat the procedure on the resulting subtree.

So far we have been discussing noncooperative game theory. Cooperative game theory deals with groups, or *coalitions* of players. When a coalition forms, the players making up the coalition draw up an agreement binding themselves to act so as to maximize the gain of the coalition. This gain is then split up among the constituents in accordance with the agreement.

1.4 Results

In Chapter 2 we prove the universality of Nash equilibria. Every real algebraic variety is isomorphic to the set of totally mixed Nash equilibria of some game with 3 players, and also of some game with N players in which each player has two pure strategies. Our proof is constructive. The numbers of pure strategies in the game with 3 players, and the number of players in the game with two pure strategies each, are polynomial in the degrees of the equations. Thus the problem of computing Nash equilibria in general is equivalent to the problem of finding the real roots of a system of polynomial equations.

In Chapter 3 we prove a theorem computing the number of solutions to a system of equations which is generic subject to the sparsity conditions embodied in a graph. We apply this theorem to games obeying graphical models and to extensive-form games. We define *emergent-node tree structures* as additional structures which normal form games may have. We apply our theorem to games having such structures. We briefly discuss how emergent node tree structures relate to cooperative games.

In Chapter 4 we discuss how to compute all Nash equilibria of a game using computer algebra. We find that polyhedral homotopy continuation is the most efficient available method in practice. It also has the advantage of being naturally parallelizable. We discuss further directions for developing algebraic algorithms for computing Nash equilibria.

Chapter 2

Universality of Nash Equilibria

2.1 Introduction

We consider the set of Nash equilibria of an N-person normal form noncooperative game with perfect information, viewed as the set of solutions to a finite system of polynomial equations and inequalities. The unknowns in this system are the components of the mixed strategy selected by each player. A set of real points given by a system of polynomial equations and inequalities is called a *semialgebraic variety*, and the special case when the system does not involve inequalities is called a *real algebraic variety*. Thus the set of Nash equilibria of a game is a semialgebraic variety.

The generic finiteness result of Harsanyi [Har73] states that for each assignment of *generic* payoffs to the normal form of a game, the number of Nash equilibria is finite and odd. In fact, McKelvey and McLennan [MM97] have computed the exact maximal number of totally mixed Nash equilibria in the generic case. Our results are complementary; they describe how complex the non-generic case can be. We show that every real algebraic variety is isomorphic (in a sense to be specified) to the set of totally mixed Nash equilibria of some game:

Theorem 1. *Every real algebraic variety is isomorphic to the set of totally mixed Nash equilibria of a 3-person game, and also of an N-person game in which each player has two pure strategies.*

The theorem of Nash [Nas52] and Tognoli [Tog73] states that every compact differentiable manifold is diffeomorphic to some (nonsingular) real algebraic variety. So, since the isomorphism above is also a diffeomorphism when the variety is nonsingular, it follows from our result that for any compact differentiable manifold M, there is some game whose set of totally mixed Nash equilibria is diffeomorphic to either M or a tubular neighborhood of M. Similarly, the theorem of Akbulut and King [AK92] shows that every piecewise linear manifold is homeomorphic to some real algebraic variety. So for every piecewise linear manifold M, there is some game whose set of totally mixed Nash equilibria is homeomorphic to either M or a tubular neighborhood of M.

Theorem 1 derives from the following more specific results:

Theorem 2. *Let $S \subset \mathbb{R}^n$ be a real algebraic variety given by m polynomial equations in n unknowns $x_1, \ldots, x_n$, such that each point $(x_1, \ldots, x_n) \in S$ satisfies $0 < x_i < 1$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} x_i < 1$. Let d be the highest power to which any unknown x_i occurs in any of the m equations. Set $D = m\left((1+d)^n - 1\right)$ and $N = nd + m$.*

(a) there is a 3-person game in which the first player has $n+1$ pure strategies, the second player has $D - m + 1$ pure strategies, and the third player has $D + 1$ pure strategies, whose set of totally mixed Nash equilibria is isomorphic to S.

(b) there is an N-person game in which each player has two pure strategies, whose set of totally mixed Nash equilibria is isomorphic to S.

Theorem 1 will follow since any real algebraic variety is isomorphic to one satisfying the hypotheses of Theorem 2. Note that specifying particular values of n, m, and d, and/or giving more detailed information about the form of the equations, may allow using games of smaller formats (fewer pure strategies in (a), fewer players in (b)). For example,

Theorem 3. *Suppose S is the set of those roots of a polynomial of degree d in one unknown which are real and lie in the open interval $(0, 1)$. Then there is a 3-person game in which the first player has two pure strategies and the other two players each have $\lceil d/2 \rceil + 1$ pure strategies,* *such that the projection of the set E of totally mixed Nash equilibria of this game onto its first component (the probability that the first player picks her first pure strategy) is S, and $\#E = \#S$.*

The notion of isomorphism being used in this chapter is that of *stable isomorphism* in the category of semialgebraic varieties. Semialgebraic varieties are the subject of study in real algebraic geometry. Two semialgebraic varieties are (semialgebraically) isomorphic if there exists a homeomorphism between them whose graph is a semialgebraic set. They are *stably isomorphic* if they are in the same equivalence class under the equivalence relation generated by semialgebraic isomorphisms and the (canonical) projections $V \times \mathbb{R}^k \to V$ for any k. Intuitively, the word "stable" here means that we are allowed to thicken the objects before mapping them isomorphically to each other.

The result in this chapter is one of a series of "universality theorems" in combinatorics. Another example is the theorem of Richter-Gebert and Ziegler [RGZ95], that realization spaces of 4-polytopes are universal. A polytope has a combinatorial description as a collection of *faces* of smaller dimensions, together with the inclusions between them (which vertices lie in which edges, etc.) The realization space of the polytope is the set of all geometric polytopes for a given combinatorial polytope, and the result states that an arbitrary primary semialgebraic set is stably equivalent to the realization space of some 4-polytope. Other universality theorems were proved by Mnëv [Mnë88] and Shor [Sho91].

2.2 Preliminaries

At a totally mixed Nash equilibrium, for any given player, if the other players' mixed strategies are kept fixed then the payoffs at each of that player's pure strategies are equal. These conditions can be expressed as a system of polynomial equations and inequalities. To simplify notation, we now give the systems for the two cases dealt with in this chapter: a 3-person game, and an N-person game in which each player has two pure strategies.

The players in the 3-person game will be called Alice, Bob, and Critter. They have $d_a + 1$, $d_b + 1$, and $d_c + 1$ pure strategies respectively. Their payoff matrices will be denoted (A_{ijk}), (B_{ijk}), and (C_{ijk}) respectively, where X_{ijk} indicates the payoff received by player X when Alice chooses her ith pure strategy, Bob chooses his jth pure strategy, and Critter chooses its kth pure strategy.

A mixed strategy of Alice is described by a d_a+1-tuple of numbers $\bar{a} = (a_0, \ldots, a_{d_a})$ with $0 \leq a_i$ for all i and $a_0 + \cdots + a_{d_a} = 1$. Here a_i signifies the fraction player A allocates to her ith pure strategy to make up this mixed strategy. Similarly, $\bar{b} = (b_0, \ldots, b_{d_b})$ and $\bar{c} = (c_0, \ldots, c_{d_c})$ denote mixed strategies of Bob and Critter, respectively. Now the expected payoff received by Alice for a particular choice $(\bar{a}, \bar{b}, \bar{c})$ of mixed strategies is

$$\sum_{i=0}^{d_a} \sum_{j=0}^{d_b} \sum_{k=0}^{d_c} A_{ijk} a_i b_j c_k,$$

and similarly for the other two players. A *totally mixed* strategy $(\bar{a}, \bar{b}, \bar{c})$ is one in which $a_i > 0$, $b_j > 0$, and $c_k > 0$ for all i, j, and k.

The condition that $(\bar{a}, \bar{b}, \bar{c})$ be a totally mixed Nash equilibrium is precisely the system of equations

$$\sum_{j=0}^{d_b} \sum_{k=0}^{d_c} A_{ijk} b_j c_k = \sum_{j=0}^{d_b} \sum_{k=0}^{d_c} A_{0jk} b_j c_k, \quad \text{for } i = 1, \ldots, d_a; \tag{2.1}$$

$$\sum_{i=0}^{d_a} \sum_{k=0}^{d_c} B_{ijk} a_i c_k = \sum_{i=0}^{d_a} \sum_{k=0}^{d_c} B_{i0k} a_i c_k, \quad \text{for } j = 1, \ldots, d_b; \tag{2.2}$$

$$\sum_{i=0}^{d_a} \sum_{j=0}^{d_b} C_{ijk} a_i b_j = \sum_{i=0}^{d_a} \sum_{j=0}^{d_b} C_{ij0} a_i b_j, \quad \text{for } k = 1, \ldots, d_c. \tag{2.3}$$

together with the inequalities $a_i > 0$, $b_j > 0$, and $c_k > 0$ for all i, j, and k.

The players in the N-person game will be denoted $X_1, \ldots, X_N$. They each have two strategies. Now a mixed strategy of player X_l is described by a pair of numbers $(p_l^{(0)}, p_l^{(1)})$, where $0 \leq p_l^{(0)}, p_l^{(1)} \leq 1$ and $p_l^{(0)} + p_l^{(1)} = 1$. Here $p_l^{(j)}$ signifies the fraction player X_l allocates to her jth pure strategy to make up this mixed strategy. We write $p_l = p_l^{(1)}$. For the strategy to be totally mixed, we must have $0 < p_l < 1$ for $l = 1, \ldots, N$. The $X_{i_1 \ldots i_N}^{(l)}$ entry of the lth player's payoff matrix indicates the payoff received by player X_l when player X_1 chooses her i_1th pure strategy, player X_2 chooses his i_2th pure strategy, and so forth. This event has probability $p_1^{(i_1)} p_2^{(i_2)} \cdots p_N^{(i_N)}$. Now holding the strategies of the other $N-1$ players fixed, one can compute the expected payoff to player X_l of choosing her first pure strategy. For example, for player X_N, this is

$$\begin{aligned} & X^{(N)}_{0\ldots 00} p^{(0)}_1 \cdots p^{(0)}_{N-2} p^{(0)}_{N-1} + X^{(N)}_{0\ldots 010} p^{(0)}_1 \cdots p^{(0)}_{N-2} p^{(1)}_{N-1} + \cdots + X^{(N)}_{1\ldots 110} p^{(1)}_1 \cdots p^{(1)}_{N-2} p^{(1)}_{N-1} \\ = \;& X^{(N)}_{0\ldots 00}(1-p_1)\cdots(1-p_{N-1}) + X^{(N)}_{0\ldots 010}(1-p_1)\cdots(1-p_{N-2})p_{N-1} + \cdots \\ & \cdots + X^{(N)}_{1\ldots 10} p_1 \cdots p_{N-1} \end{aligned}$$

Observe that this expression is multilinear in the p_l's and that p_N does not occur (because we are conditioning on the event that X_N chooses her first pure strategy). At a totally mixed Nash equilibrium, this must equal the expected payoff to player X_N of choosing her second pure strategy. In this way, a totally mixed Nash equilibrium corresponds to a solution to a system of N multilinear equations, where the lth unknown p_l does not occur in the lth equation:

$$\begin{aligned} & X^{(1)}_{00\ldots 0}(1-p_2)\cdots(1-p_N) + X^{(1)}_{00\ldots 01}(1-p_2)\cdots(1-p_{N-1})p_N + \cdots \\ & \quad \cdots + X^{(1)}_{01\ldots 1} p_2 \cdots p_N \\ = \;& X^{(1)}_{10\ldots 0}(1-p_2)\cdots(1-p_N) + X^{(1)}_{10\ldots 01}(1-p_2)\cdots(1-p_{N-1})p_N + \cdots \\ & \quad \cdots + X^{(1)}_{11\ldots 1} p_2 \cdots p_N, \end{aligned}$$

$$\vdots$$

$$\begin{aligned} & X^{(N)}_{0\ldots 00}(1-p_1)\cdots(1-p_{N-1}) + X^{(N)}_{0\ldots 010}(1-p_1)\cdots(1-p_{N-2})p_{N-1} + \cdots \\ & \quad \cdots + X^{(N)}_{1\ldots 10} p_1 \cdots p_{N-1} \\ = \;& X^{(N)}_{0\ldots 01}(1-p_1)\cdots(1-p_{N-1}) + X^{(N)}_{0\ldots 011}(1-p_1)\cdots(1-p_{N-2})p_{N-1} + \cdots \\ & \quad \cdots + X^{(N)}_{1\ldots 11} p_1 \cdots p_{N-1}. \end{aligned}$$

See for example McKelvey and McLennan [MM96] or Sturmfels [Stu02], Chapter 6.

2.3 Examples

We illustrate our method of encoding real varieties with a few numerical examples. In the first two examples, the set of totally mixed Nash equilibria is isomorphic to a circle. In the last example, there is a unique totally mixed Nash equilibrium, and its degree is 2. The same method allows us to construct a game with a unique totally mixed Nash equilibrium of any given topological degree.

We define a 3-person game in which each player has three pure strategies as follows. If a player picks their first pure strategy, then all that player's payoffs are zero. If Alice picks her second pure strategy, then her payoff matrix is

$$(A_{1jk}) = \begin{array}{c} \\ b_0 \\ b_1 \\ b_2 \end{array} \begin{array}{c} \begin{array}{ccc} c_0 & c_1 & c_2 \end{array} \\ \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \end{array}. \tag{2.4}$$

If she picks her third pure strategy, her payoff is always zero. If Bob picks his second pure strategy, his payoff matrix is

$$(B_{i1k}) = \begin{array}{c} \\ a_0 \\ a_1 \\ a_2 \end{array} \begin{array}{c} \begin{array}{ccc} c_0 & c_1 & c_2 \end{array} \\ \begin{pmatrix} \frac{-1}{2} & \frac{-1}{2} & \frac{1}{2} \\ \frac{-1}{2} & \frac{-1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{pmatrix} \end{array}, \tag{2.5}$$

and if he picks his third pure strategy, his payoff matrix is

$$(B_{i2k}) = \begin{array}{c} \\ a_0 \\ a_1 \\ a_2 \end{array} \begin{array}{c} \begin{array}{ccc} c_0 & c_1 & c_2 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \end{array}. \tag{2.6}$$

If Critter picks its second pure strategy, its payoff matrix is

$$(C_{ij1}) = \begin{array}{c} \\ a_0 \\ a_1 \\ a_2 \end{array} \begin{array}{c} \begin{array}{ccc} b_0 & b_1 & b_2 \end{array} \\ \begin{pmatrix} \frac{-1}{2} & \frac{-1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \\ \frac{-1}{2} & \frac{-1}{2} & \frac{1}{2} \end{pmatrix} \end{array}, \tag{2.7}$$

and if it picks its third pure strategy, its payoff matrix is

$$(C_{ij2}) = \begin{matrix} & b_0 & b_1 & b_2 \\ a_0 \\ a_1 \\ a_2 \end{matrix} \begin{pmatrix} \frac{7}{64} & \frac{-57}{64} & \frac{7}{64} \\ \frac{7}{64} & \frac{-57}{64} & \frac{-57}{64} \\ \frac{7}{64} & \frac{-57}{64} & \frac{7}{64} \end{pmatrix}. \tag{2.8}$$

Suppose $(\bar{a}, \bar{b}, \bar{c})$ is a totally mixed Nash equilibrium. Then in particular, Alice's payoff from her second pure strategy, which from equation (2.4) is

$$\begin{aligned} -b_0c_1 + b_1c_0 + b_1c_2 - b_2c_1 &= -(1 - b_1 - b_2)c_1 + b_1(1 - c_1 - c_2) + b_1c_2 - b_2c_2 \\ &= b_1 - c_1, \end{aligned}$$

must be equal to her payoff from her first pure strategy, which is zero. In this way one can verify that when a totally mixed Nash equilibrium occurs, $b_1 = c_1$, $a_2 + c_2 - \frac{1}{2} = 0$, $c_1 - a_2c_2 = 0$, $a_1 + b_2 - \frac{1}{2} = 0$, and $-a_1b_2 - b_1 + \frac{7}{64} = 0$. Substituting into the last equation yields

$$a_1\left(a_1 - \frac{1}{2}\right) + a_2\left(a_2 - \frac{1}{2}\right) + \frac{7}{64} = 0$$

which can be rewritten as

$$\left(a_1 - \frac{1}{4}\right)^2 + \left(a_2 - \frac{1}{4}\right)^2 = \frac{1}{64}.$$

This equation describes a circle in the a_1a_2 plane, which lies completely in the interior of the coordinate simplex $0 < a_1$, $0 < a_2$, $a_1 + a_2 < 1$. One can check that for these values of (a_1, a_2), the values of (b_1, b_2) and (c_1, c_2) given by the above equations also lie in the interiors of their respective coordinate simplices. So the set of totally mixed Nash equilibria is exactly the set of solutions to the above equations, which is isomorphic to a circle.

We can obtain the same circle as the set of totally mixed Nash equilibria of a 5-person game in which each player has two pure strategies. We will call the players Alice, Bob, Critter, Deus, and Elizabeth, with payoff matrices A, B, C, D, and E respectively. Alice's mixed strategy is denoted $(a_0, a_1) = (1 - a, a)$, and likewise for the other players. If a player picks their first pure strategy, then all that player's payoffs are zero. If Alice picks her second pure strategy, her payoff depends only on what Bob and Deus do. Her payoff matrix is

$$(A_{1i\bullet j\bullet}) = \begin{matrix} & d_0 & d_1 \\ b_0 & 0 & -1 \\ b_1 & 1 & 0 \end{matrix}. \tag{2.9}$$

If Bob picks his second pure strategy, his payoff depends only on what Alice and Critter do. His payoff matrix is

$$(B_{i1j\bullet\bullet}) = \begin{matrix} & c_0 & c_1 \\ a_0 & 0 & -1 \\ a_1 & 1 & 0 \end{matrix}. \tag{2.10}$$

If Critter picks its second pure strategy, its payoff depends only on what Elizabeth does. Its payoff matrix is

$$(C_{\bullet\bullet 1\bullet i}) = \begin{matrix} e_0 & e_1 \\ -\frac{1}{2} & \frac{1}{2} \end{matrix}. \tag{2.11}$$

Deus's payoff is always zero. Finally, if Elizabeth picks her second pure strategy, her payoff is given by

$$E_{ijkl1} = \begin{matrix} & c_0d_0 & c_0d_1 & c_1d_0 & c_1d_1 \\ a_0b_0 & \frac{7}{64} & \frac{7}{64} & \frac{7}{64} & \frac{7}{64} \\ a_0b_1 & \frac{-25}{64} & \frac{39}{64} & \frac{-25}{64} & \frac{39}{64} \\ a_1b_0 & \frac{-25}{64} & \frac{-25}{64} & \frac{39}{64} & \frac{39}{64} \\ a_1b_1 & \frac{-57}{64} & \frac{7}{64} & \frac{7}{64} & \frac{71}{64} \end{matrix} \tag{2.12}$$

One can verify that when a totally mixed Nash equilibrium occurs, $b-d=0$, $a-c=0$, $e-\frac{1}{2}=0$, and $ac-\frac{1}{2}a+bd-\frac{1}{2}b+\frac{7}{64}=0$. Substituting into this last equation yields

$$a^2 - \frac{1}{2}a + b^2 - \frac{1}{2}b + \frac{7}{64} = 0,$$

the same equation as before. It is clear that for a and b satisfying this equation, the values of c, d, and e given by the above equations satisfy $0 < c, d, e < 1$. So again the set of totally mixed Nash equilibria is isomorphic to a circle.

We next define another 5-person game in which each player has two pure strategies. Again, if a player picks their first pure strategy, then all that player's payoffs are zero. Alice's, Bob's, and Critter's payoffs are the same as in the previous example. Now Deus's payoff for picking his second pure strategy depends only on what Alice and Bob do, and his payoff matrix is

$$(D_{ij\bullet 1\bullet}) = \begin{array}{c} \\ a_0 \\ a_1 \end{array} \begin{array}{c} \begin{matrix} b_0 & b_1 \end{matrix} \\ \begin{pmatrix} \frac{1}{8} & \frac{-3}{8} \\ \frac{-3}{8} & \frac{9}{8} \end{pmatrix} \end{array}. \tag{2.13}$$

This time if Elizabeth picks her second pure strategy, her payoff is given by

$$E_{ijkl1} = \begin{array}{c} \\ a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{array} \begin{array}{c} \begin{matrix} c_0 d_0 & c_0 d_1 & c_1 d_0 & c_1 d_1 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & -1 & 1 & 0 \end{pmatrix} \end{array} \tag{2.14}$$

Now when a totally mixed Nash equilibrium occurs, $b - d = 0$, $a - c = 0$, and $e - \frac{1}{2} = 0$ as before, and one can verify that $2ab - \frac{1}{2}a - \frac{1}{2}b + \frac{1}{8} = 0$ and $ac - bd - \frac{1}{2}a + \frac{1}{2}b = 0$ as well. The fourth equation may be rewritten as $2(a - \frac{1}{4})(b - \frac{1}{4}) = 0$. Substituting the first two equations into the fifth yields $a^2 - \frac{1}{2}a - b^2 + \frac{1}{2}b = 0$, which may be rewritten as $(a - \frac{1}{4})^2 - (b - \frac{1}{4})^2 = 0$. Clearly the unique totally mixed Nash equilibrium occurs when $a = \frac{1}{4}$, $b = \frac{1}{4}$. Putting $x = a - \frac{1}{4}$ and $y = b - \frac{1}{4}$, we see that the polynomials defining the totally mixed Nash equilibrium are $(x^2 - y^2, 2xy)$, which is the canonical example of a map from the plane to itself of degree 2; considering the plane as having complex coordinate $z = x + iy$, it is given by $z \mapsto z^2$. Every map $z \mapsto z^n$ for $n \in \mathbb{N}$ similarly gives rise to a system of two polynomial equations in two variables, and so we can obtain a unique totally mixed Nash equilibrium of any given topological degree.

2.4 Proofs

We begin with some simple observations. In the equations (2.1) for Alice to be at a totally mixed Nash equilibrium in the 3-player case, if we substitute $b_0 = 1 - b_1 - \cdots - b_{d_b}$ and $c_0 = 1 - c_1 - \cdots - c_{d_c}$, and subtract each right-hand side from each left-hand side, we get d_a equations

$$\lambda^{(i)}_{11} b_1 c_1 + \lambda^{(i)}_{12} b_1 c_2 + \cdots \lambda^{(i)}_{1d_c} b_1 c_{d_c} + \lambda^{(i)}_{21} b_2 c_1 + \cdots + \lambda^{(i)}_{(d_b)(d_c)} b_{d_b} c_{d_c}$$
$$+\lambda^{(i)}_{10} b_1 + \cdots \lambda^{(i)}_{(d_b)0} b_{d_b} + \lambda^{(i)}_{01} c_1 + \cdots + \lambda^{(i)}_{0(d_c)} c_{d_c} + \lambda^{(i)}_{00} = 0 \quad (A)$$

for $i = 1, \ldots, d_a$. Similarly we get d_b equations

$$\mu^{(j)}_{11} a_1 c_1 + \mu^{(j)}_{12} a_1 c_2 + \cdots \mu^{(j)}_{1d_c} a_1 c_{d_c} + \mu^{(j)}_{21} a_2 c_1 + \cdots + \mu^{(j)}_{(d_a)(d_c)} a_{d_a} c_{d_c}$$
$$+\mu^{(j)}_{10} a_1 + \cdots \mu^{(j)}_{(d_a)0} a_{d_a} + \mu^{(j)}_{01} c_1 + \cdots + \mu^{(j)}_{0(d_c)} c_{d_c} + \mu^{(j)}_{00} = 0 \quad (B)$$

for $j = 1, \ldots, d_b$, and d_c equations

$$\nu^{(k)}_{11} a_1 b_1 + \nu^{(k)}_{12} a_1 b_2 + \cdots \nu^{(k)}_{1d_b} a_1 b_{d_b} + \nu^{(k)}_{21} a_2 b_1 + \cdots + \nu^{(k)}_{(d_a)(d_b)} a_{d_a} b_{d_b}$$
$$+\nu^{(k)}_{10} a_1 + \cdots \nu^{(k)}_{(d_a)0} a_{d_a} + \nu^{(k)}_{01} b_1 + \cdots + \nu^{(k)}_{0(d_b)} b_{d_b} + \nu^{(k)}_{00} = 0 \quad (C)$$

for $k = 1, \ldots, d_c$.

Lemma 4. *If we are given any arbitrary coefficients $\lambda^{(i)}_{jk}$, $\mu^{(j)}_{ik}$, and $\nu^{(k)}_{ij}$, we can choose payoff matrices (A_{ijk}), (B_{ijk}), and (C_{ijk}) so that the above equations have the prescribed coefficients.*

Proof. We first set each player's payoff equal to zero whenever they choose their first pure strategy, no matter what the other players do. Then the equations (2.1) imply that, for example, $\sum_{j=1}^{d_b} \sum_{k=1}^{d_c} A_{1jk} b_j c_k = 0$. Now we show how to obtain the first of the above equations,

$$\lambda^{(1)}_{11} b_1 c_1 + \lambda^{(1)}_{12} b_1 c_2 + \cdots \lambda^{(1)}_{1d_c} b_1 c_{d_c} + \lambda^{(1)}_{21} b_2 c_1 + \cdots + \lambda^{(1)}_{(d_b)(d_c)} b_{d_b} c_{d_c}$$
$$+\lambda^{(1)}_{10} b_1 + \cdots \lambda^{(1)}_{(d_b)0} b_{d_b} + \lambda^{(1)}_{01} c_1 + \cdots + \lambda^{(1)}_{0(d_c)} c_{d_c} + \lambda^{(1)}_{00} = 0,$$

for any prescribed $\lambda^{(1)}_{jk}$ by appropriately choosing the matrix (A_{1jk}); the other equations are obtained completely analogously. First, we set $A_{100} = \lambda^{(1)}_{00}$. Then for $j =$

$1, \ldots, d_b$, we set $A_{1j0} = \lambda_{j0}^{(1)} + \lambda_{00}^{(1)}$, and for $k = 1, \ldots, d_c - 1$, we set $A_{10k} = \lambda_{0k}^{(1)} + \lambda_{00}^{(1)}$. Finally, for $j = 1, \ldots, d_b$ and $k = 1, \ldots, d_c$, we set $A_{1jk} = \lambda_{jk}^{(1)} + \lambda_{j0}^{(1)} + \lambda_{0k}^{(1)} + \lambda_{00}^{(1)}$. This yields

$$\begin{aligned}\sum_{j=0}^{d_b} \sum_{0=1}^{d_c} A_{1jk} b_j c_k = \lambda_{00}^{(1)} \left(1 - \sum_{j=1}^{d_b} b_j\right) \left(1 - \sum_{k=1}^{d_c} c_k\right) \\ + \sum_{j=1}^{d_b} \left(\lambda_{j0}^{(1)} + \lambda_{00}^{(1)}\right) b_j \left(1 - \sum_{k=1}^{d_c} c_k\right) \\ +(1 - \sum_{j=1}^{d_b} b_j) \sum_{k=1}^{d_c} (\lambda_{0k}^{(1)} + \lambda_{00}^{(1)}) c_k \\ + \sum_{j=1}^{d_b} \sum_{k=1}^{d_c} (\lambda_{jk}^{(1)} + \lambda_{j0}^{(1)} + \lambda_{0k}^{(1)} + \lambda_{00}^{(1)}) b_j c_k \\ = \lambda_{00}^{(1)} + \sum_{j=1}^{d_b} \lambda_{j0}^{(1)} b_j + \sum_{k=1}^{d_c} \lambda_{0k}^{(1)} c_k + \sum_{j=1}^{d_b} \sum_{k=1}^{d_c} \lambda_{jk}^{(1)} b_j c_k\end{aligned}$$

as desired. □

Similarly, for the N-person game we have N equations, one for each player l, where the lth equation is of the form

$$\sum_{\varepsilon \in \{0,1\}^{N-1}} \lambda_{\varepsilon}^{(l)} p_1^{\varepsilon_1} \cdots p_{l-1}^{\varepsilon_{l-1}} p_{l+1}^{\varepsilon_l} \cdots p_N^{\varepsilon_{N-1}} = 0.$$

Lemma 5. *If we are given any arbitrary coefficients* $\lambda_{\varepsilon}^{(l)}$*, we can choose payoff matrices* $X_{i_1 \ldots i_N}^{(l)}$ *so that the above equations have the prescribed coefficients.*

Proof. As before, we set all players' payoffs from their first pure strategies equal to zero, regardless of what the other players do. We show how to obtain the equation

$$\sum_{\varepsilon \in \{0,1\}^{N-1}} \lambda_{\varepsilon}^{(N)} p_1^{\varepsilon_1} \cdots p_{N-1}^{\varepsilon_{N-1}} = 0$$

for any given $\lambda_{\varepsilon}^{(N)}$ by appropriately choosing payoff $X_{i_1 \ldots i_{N-1} 1}^{(N)}$; again, the other equations are obtained analogously. We set

$$X_{i_1 \ldots i_{N-1} 1}^{(N)} = \sum_{\varepsilon_1 = 0}^{i_1} \sum_{\varepsilon_2 = 0}^{i_2} \cdots \sum_{\varepsilon_{N-1} = 0}^{i_{N-1}} \lambda_{\varepsilon}^{(N)}.$$

We show by induction on N that

$$X^{(N)}_{0\ldots01}(1-p_1)\cdots(1-p_{N-1})+\cdots X^{(N)}_{1\ldots11}p_1\cdots p_{N-1}=\sum_{\varepsilon\in\{0,1\}^{N-1}}\lambda^{(N)}_{\varepsilon}p_1^{\varepsilon_1}\cdots p_{N-1}^{\varepsilon_{N-1}}$$

For $N=2$, we have

$$X^{(2)}_{01}(1-p_1)+X^{(2)}_{11}p_1=\lambda^{(2)}_0(1-p_1)+(\lambda^{(2)}_0+\lambda^{(2)}_1)p_1=\lambda^{(2)}_0+\lambda^{(2)}_1p_1$$

as desired. Now suppose we have shown the identity for $N-1$. We observe that

$$X^{(N)}_{i_1\ldots i_{N-2}01}=\sum_{\varepsilon_1=0}^{i_1}\sum_{\varepsilon_2=0}^{i_2}\cdots\sum_{\varepsilon_{N-2}=0}^{i_{N-2}}\lambda^{(N)}_{\varepsilon_1\ldots\varepsilon_{N-2}0}.$$

and

$$X^{(N)}_{i_1\ldots i_{N-2}11}=\sum_{\varepsilon_1=0}^{i_1}\sum_{\varepsilon_2=0}^{i_2}\cdots\sum_{\varepsilon_{N-2}=0}^{i_{N-2}}(\lambda^{(N)}_{\varepsilon_1\ldots\varepsilon_{N-2}0}+\lambda^{(N)}_{\varepsilon_1\ldots\varepsilon_{N-2}1})$$

Then

$$\begin{aligned}
&X^{(N)}_{0\ldots01}(1-p_1)\cdots(1-p_{N-1})+\cdots X^{(N)}_{1\ldots11}p_1\cdots p_{N-1}\\
=\;&\left(X^{(N)}_{0\ldots001}(1-p_1)\cdots(1-p_{N-2})+\cdots+X^{(N)}_{1\ldots101}p_1\cdots p_{N-2}\right)(1-p_{N-1})\\
&+\Big(X^{(N)}_{0\ldots011}(1-p_1)\cdots(1-p_{N-2})+\cdots\\
&\qquad\cdots+X^{(N)}_{1\ldots111}p_1\cdots p_{N-2}\Big)p_{N-1}\\
=\;&\left(\sum_{\varepsilon'\in\{0,1\}^{N-2}}\lambda^{(N)}_{\varepsilon'0}p_1^{\varepsilon'_1}\cdots p_{N-2}^{\varepsilon'_{N-2}}\right)(1-p_{N-1})\\
&+\left(\sum_{\varepsilon'\in\{0,1\}^{N-2}}(\lambda^{(N)}_{\varepsilon'0}+\lambda^{(N)}_{\varepsilon'1})p_1^{\varepsilon'_1}\cdots p_{N-2}^{\varepsilon'_{N-2}}\right)p_{N-1}\\
=\;&\sum_{\varepsilon'\in\{0,1\}^{N-2}}(\lambda^{(N)}_{\varepsilon'0}+\lambda^{(N)}_{\varepsilon'1}p_{N-1})p_1^{\varepsilon'_1}\cdots p_{N-2}^{\varepsilon'_{N-2}}\\
=\;&\sum_{\varepsilon\in\{0,1\}^{N-1}}\lambda^{(N)}_{\varepsilon}p_1^{\varepsilon_1}\cdots p_{N-1}^{\varepsilon_{N-1}}
\end{aligned}$$

□

In fact Lemma 4 and Lemma 5 are instances of a more general lemma, which is also proved in [Dat03a].

This lemma, which also appears for example in McKelvey and McLennan [MM97] or Sturmfels [Stu02], Chapter 6, is crucial to the proofs of the following theorems, from which Theorem 2 follows:

Theorem 6. *Let $S \subset \mathbb{R}^n$ be a real algebraic variety given by m equations in n unknowns $x_1, \ldots, x_n$, such that each point $(x_1, \ldots, x_n) \in S$ satisfies $0 < x_i$ for $i = 1, \ldots, n$ and $\sum_{i=1}^n x_i < 1$, and suppose the highest power of x_i in equation j is $x_i^{d_{ij}}$. Set*

$$D = -1 + \sum_{j=1}^{m} (1 + d_{1j})(1 + d_{2j}) \cdots (1 + d_{nj}).$$

Then there is a 3-person game in which Alice has $n + 1$ pure strategies, Bob has $D - m + 1$ pure strategies, and Critter has $D + 1$ pure strategies, whose set of totally mixed Nash equilibria is isomorphic to S.

Proof. We suppose S to be given by the m equations

$$F_j(x_1, \ldots, x_n) = 0$$

for $j = 1, \ldots, m$. We now consider the equations (C) associated with Critter's payoffs; recall that these are equations involving only the a_i's and b_i's. We will show how an arbitrary system of polynomial equations $F_j(x_1, \ldots, x_n) = 0$ can be encoded in this system. (We will consider the c_i's and the equations (A) and (B) later.) The variables $a_1, \ldots, a_n$ will take the roles of $x_1, \ldots, x_n$.

We will repeatedly use the following observation. Suppose we have a system of polynomial equations

$$\begin{aligned} f_1(x_1, \ldots, x_i) &= 0, \\ &\vdots \\ f_k(x_1, \ldots, x_i) &= 0 \end{aligned}$$

such that $f_k(x_1, \ldots, x_i) = \alpha x_i + g(x_1, \ldots, x_{i-1})$ where α is some nonzero constant coefficient and g is a polynomial in the remaining variables other than x_i. Then our system is logically equivalent to (i.e., it implies and is implied by) the system

$$
\begin{aligned}
f_1(x_1, \ldots, x_{i-1}, -\alpha^{-1} g(x_1, \ldots, x_{i-1},)) &= 0, \\
&\vdots, \\
f_{k-1}(x_1, \ldots, x_{i-1}, -\alpha^{-1} g(x_1, \ldots, x_{i-1})) &= 0, \\
x_i &= -\alpha^{-1} g(x_1, \ldots, x_{i-1}).
\end{aligned}
$$

Effectively we have substituted the value of x_i given by the last equation into the other equations. Notice that the variable x_i no longer appears in the first $i-1$ equations. In our construction we will actually be going the other way: we will be starting with a system of equations in fewer variables and adding a new variable x_i as above. The old system defined a variety V lying in $\mathbb{R}^{i-1}$, and the new system defines a variety V' lying in $\mathbb{R}^i$. The two varieties are isomorphic, with isomorphism given by the embedding $(x_1, \ldots, x_{i-1}) \mapsto (x_1, \ldots, x_{i-1}, -\alpha^{-1} g(x_1, \ldots, x_{i-1}))$.

Most of the equations in our system will be of the form

$$b'_i = \lambda a_l b'_j + \lambda'$$

for some constants λ and λ', where $b'_i = s_i b_i + \delta_i$ for some constants s_i and δ_i. We imagine that we are computing with a device which allows us to multiply a previous result b'_j by one of the coordinates a_l and a constant λ, add another constant λ', and store the result in b'_i. By multiplying and adding in this way, we will eventually be able to evaluate the polynomial $F_j(a_1, \ldots, a_n)$, and then we finally use another equation to express the constraint that $F_j(a_1, \ldots, a_n) = 0$. The s_i's and δ_i's are constants with $s_i \neq 0$, which we choose so that for any point $(a_1, \ldots, a_n) \in S$, we will have $0 < b_i$ for all i and $\sum_{i=1}^{D-m} b_i < 1$. This is possible since the set S fits inside $(0, 1)^n$.

Write F_j in recursive form as

$$
\begin{aligned}
F_j(x_1, \ldots, x_n) &= x_1^{d_{1j}} F_{jd_{1j}}(x_2, \ldots, x_n) + \cdots + F_{j0}(x_2, \ldots, x_n) \\
&= \cdots = \\
&= x_1^{d_{1j}}(x_2^{d_{2j}} \cdots (x_n^{d_{nj}} F_{jd_{1j}\ldots d_{nj}} + \cdots) \cdots) + \cdots + F_{j0\ldots0}
\end{aligned}
$$

where the $F_{ji_1\ldots i_n}$ are constants, the $F_{ji_1\ldots i_{n-1}}$ are polynomials in a_n, the $F_{ji_1\ldots i_{n-2}}$ are polynomials in a_{n-1} and a_n, and so forth. To evaluate $F_j(a_1, \ldots, a_n)$, first we will evaluate the polynomials $F_{ji_1\ldots i_{n-1}}(a_n)$. Then we will evaluate the polynomials $F_{ji_1\ldots i_{n-2}}(a_{n-1}, a_n)$, which are polynomials in a_{n-1} with the $F_{ji_1\ldots i_{n-1}}(a_n)$ we computed previously as coefficients; and so forth.

At each stage, we will be evaluating a polynomial in *one* of the a_i's, whose coefficients are some of our previous results. To evaluate this univariate polynomial, we will use Horner's rule, which states that a univariate polynomial

$$\xi_d x^d + \xi_{d-1}x^{d-1} + \cdots + \xi_1 x + \xi_0$$

can be evaluated as

$$(\cdots((\xi_d x + \xi_{d-1})x + \xi_{d-2})x + \cdots + \xi_1)x + \xi_0.$$

Our first equation is

$$s_1 b_1 + \delta_1 = a_n F_{1d_{11}\ldots d_{n1}} + F_{1d_{11}\ldots d_{(n-1)1}(d_{n1}-1)}.$$

Our second equation is

$$s_2 b_2 + \delta_2 = a_n(s_1 b_1 + \delta_1) + F_{1d_{11}\ldots d_{(n-1)1}(d_{n1}-2)}.$$

Continuing in this way, our d_{n1}th equation is

$$s_{d_{n1}} b_{d_{n1}} + \delta_{d_{n1}} = a_n(s_{(d_{n1}-1)} b_{(d_{n1}-1)} + \delta_{(d_{n1}-1)}) + F_{1d_{11}\ldots d_{(n-1)1}0}.$$

Observe that the righthand side of this last equation is $F_{1d_{11}\ldots d_{(n-1)1}}(a_n)$. In the same way, we obtain all the polynomials $F_{1i_1\ldots i_{n-1}}(a_n)$ for $i_1 = 0, \ldots, d_{11}, \ldots, i_{n-1} = 0, \ldots, d_{(n-1)1}$, setting up d_{n1} equations for each. This takes care of the first $k = (1 + d_{11})(1 + d_{21}) \cdots (1 + d_{(n-1)1}) d_{n1}$ equations. Now we start building up the bivariate polynomials. We begin by constructing $d_{(n-1)1}$ equations starting with

$$s_{k+1} b_{k+1} + \delta_{k+1} = a_{n-1}(s_{d_{n1}} b_{d_{n1}} + \delta_{d_{n1}}) + (s_{2d_{n1}} b_{2d_{n1}} + \delta_{2d_{n1}}),$$

and end up with $F_{1d_{11}\dots d_{(n-2)1}}(a_{n-1}, a_n)$ on the righthand side. In this way we use $(1+d_{11})(1+d_{21})\cdots(1+d_{(n-2)1})d_{(n-1)1}$ more equations to obtain all the polynomials $F_{1i_1\dots i_{n-2}}(a_{n-1}, a_n)$ for $i_1 = 0, \dots, d_{11}, \dots, i_{n-2} = 0, \dots, d_{(n-2)1}$. Continuing in this manner, we at last end up with the equation $0 = F_1(a_1, \dots, a_n)$. We have used

$$\begin{aligned} & d_{11} + (1+d_{11})d_{21} + \cdots + (1+d_{11})(1+d_{21})\cdots(1+d_{(n-1)1})d_{n1} \\ = \; & (1+d_{11})(1+d_{21})\cdots(1+d_{n1}) - 1 \end{aligned}$$

equations. In this way we construct D equations to encode all the m equations $0 = F_j(a_1, \dots, a_n)$. The lefthand sides of each of these equations contains a distinct b_i, except for the m equations $0 = F_j(a_1, \dots, a_n)$ themselves. Thus we have made the set of totally mixed Nash equilibria consist exactly of those points $(a_1, \dots, a_n)$ in the set S, and for each such point we have set the values of all $D - m + 1$ b_i's (the last equation is $\sum b_i = 1$).

It remains to set the values of the D c_i's. We have n equations (A) and $D - m$ equations (B) left, each of which we can use to set some c_i equal to $\frac{1}{D}$. If $m > n$ there will be $m - n$ c_i's left over. These are unconstrained except that $0 < c_i < 1$ and $\sum c_i = 1$. Thus the set of totally mixed Nash equilibria will be a Cartesian product of S and a product of open simplices, which is stably isomorphic to S. □

Theorem 7. *Let $S \subset \mathbb{R}^n$ be a real algebraic variety given by m equations in n unknowns $x_1, \dots, x_n$, such that each point $(x_1, \dots, x_n) \in S$ satisfies $0 < x_i$ for $i = 1, \dots, n$ and $\sum_{i=1}^n x_i < 1$, and suppose the highest power of x_i in equation j is $x_i^{d_{ij}}$. Set*

$$D' = \sum_{i=1}^{n} \max_j d_{ij}.$$

Then there is a game with $(D' + m)$ players in which each player has 2 pure strategies, whose set of totally mixed Nash equilibria is isomorphic to S.

Proof. We first give a game with $D' + \max\{m, n\}$ players. We take the first n variables $p_1, \dots, p_n$ to represent $x_1, \dots, x_n$. Let $d_i = \max_j d_{ij}$, and rename the last D' variables as $p_{11}, \ldots, p_{1d_1}, \ldots, p_{n1}, \ldots, p_{nd_n}$. Then the last D' equations are

$$\begin{array}{rcl} p_{11} & = & p_1, \quad p_{12} = p_1 p_{11}, \quad \ldots, \quad p_{1d_1} = p_1 p_{1(d_1-1)}; \\ \vdots & & \\ p_{n1} & = & p_n, \quad p_{n2} = p_n p_{n1}, \quad \ldots, \quad p_{nd_n} = p_n p_{n(d_n-1)}. \end{array}$$

The first m equations are $F_j(x_1, \quad \ldots, \quad x_n) = 0$ for $j = 1, \ldots, m$, with x_i^k replaced by p_{ik}. Any remaining equations can be $0 = 0$.

Note that this means the first n variables $p_1, \ldots, p_n$ do not occur in the first n equations. If $m > n$, the next $m - n$ variables do not occur in any equations. We must arrange the last D' equations such that p_{ij} does not occur in the (i, j)th equation. We could show that such an arrangement exists using Philip Hall's Marriage Theorem, but for concreteness we instead construct one such arrangement explicitly. If $d_i \geq 3$, we arrange the equations involving p_i as follows:

$$\begin{array}{lrcl} (i, 1) & p_{i3} & = & p_i p_{i2}, \\ & & \vdots & \\ (i, d_i - 2) & p_{id_i} & = & p_i p_{i(d_i-1)}, \\ (i, d_i - 1) & p_{i1} & = & p_i, \\ (i, d_i) & p_{i2} & = & p_i p_{i1}. \end{array}$$

So it remains to consider those i for which $d_i = 1$ or $d_i = 2$. To simplify notation, assume that $d_i = 1$ for $i = 1, \ldots, u$, that $d_i = 2$ for $i = u + 1, \ldots, u + v$, and that $d_i > 2$ for $i > u + v$.

If $u = v = 0$, we're done. If $u \geq 1$ and $v \geq 1$, then we can arrange the remaining equations as follows:

$$\begin{array}{llcl}
(1,1) & p_{21} & = & p_2, \\
 & & \vdots & \\
(u-1,1) & p_{u1} & = & p_u, \\
(u,1) & p_{(u+1)2} & = & p_{(u+1)}p_{(u+1)1}, \\
(u+1,1) & p_{(u+2)1} & = & p_{(u+2)}, \\
(u+1,2) & p_{(u+2)2} & = & p_{(u+2)}p_{(u+2)1}, \\
 & & \vdots & \\
(u+v-1,1) & p_{(u+v)1} & = & p_{(u+v)}, \\
(u+v-1,2) & p_{(u+v)2} & = & p_{(u+v)}p_{(u+v)1}, \\
(u+v,1) & p_{11} & = & p_1, \\
(u+v,2) & p_{(u+1)1} & = & p_{(u+1)}.
\end{array}$$

If $v = 0$ and $u \geq 2$, we instead arrange the u remaining equations as follows:

$$\begin{array}{llcl}
(1,1) & p_{21} & = & p_2, \\
 & & \vdots & \\
(u-1,1) & p_{u1} & = & p_u, \\
(u,1) & p_{11} & = & p_1.
\end{array}$$

If $u = 0$ and $v \geq 2$, we instead arrange the v remaining equations as follows:

$$\begin{array}{llcl}
(1,1) & p_{21} & = & p_2, \\
(1,2) & p_{22} & = & p_2p_{21}, \\
 & & \vdots & \\
(v-1,1) & p_{v1} & = & p_v, \\
(v-1,2) & p_{v2} & = & p_vp_{v1}, \\
(v,1) & p_{11} & = & p_1, \\
(v,2) & p_{12} & = & p_1p_{11}.
\end{array}$$

If $u = 1$ and $v = 0$, we do not actually need the equation $p_{11} = p_1$. Recall that we had replaced x_1 by p_{11} in the first m equations, $F_j(x_1, \ldots, x_n) = 0$ for $j = 1, \ldots, m$. Instead, we put $p_{11} = \frac{1}{2}$ for the very first equation, replace x_1 by p_1 in the next $m-1$

equations $F_2(x_1, \ldots, x_n) = 0, \ldots, F_m(x_1, \ldots, x_n) = 0$, and also replace x_1 by p_1 in the $(1, 1)$th equation $F_1(x_1, \ldots, x_n) = 0$.

It remains to consider the case where $u = 0, v = 1$. If there is at least one more variable, then we have $d_2 \geq 3$, so the first $d_2 + 2$ of the D' remaining equations can be

$$
\begin{array}{lrcl}
(1,1) & p_{21} & = & p_2, \\
(1,2) & p_{22} & = & p_2 p_{21}, \\
(2,1) & p_{23} & = & p_2 p_{22}, \\
 & & \vdots & \\
(2, d_2-2) & p_{2d_2} & = & p_2 p_{2(d_2-1)}, \\
(2, d_2-1) & p_{11} & = & p_1, \\
(2, d_2) & p_{12} & = & p_1 p_{11}.
\end{array}
$$

So now suppose there is only one variable $x = x_1$. We have $n = 1$ and $D' = 2$; our game has $m + 2$ players. Write the polynomials $F_j(x_1)$ as $F_j(x_1) = a_j x_1^2 + b_j x_1 + c_j$ for $j = 1, \ldots, m$. Then the $m + 2$ equations are

$$
\begin{array}{lrcl}
(1) & a_1 p_{(m+1)} p_{(m+2)} + b_1 p_{(m+1)} + c_1 & = & 0, \\
(2) & a_2 p_1 p_{(m+1)} + b_2 p_1 + c_2 & = & 0, \\
 & & \vdots & \\
(m) & a_m p_1 p_{(m+1)} + b_m p_1 + c_m & = & 0, \\
(m+1) & p_{(m+2)} & = & p_1, \\
(m+2) & p_{(m+1)} & = & p_1.
\end{array}
$$

Now we have encoded S in a game with $D' + \max\{m, n\}$ players. It remains to show that we only need $D' + m$ players. So suppose $n > m$. Above, we have used d_i equations $p_{i1} = p_i, \ldots, p_{id_i} = p_i p_{i(d_i-1)}$, but we could have gotten away with $d_i - 1$ equations instead if we were willing to use p_i in the equations encoding the polynomials $F_j(x_1, \ldots, x_n) = 0$. In that case all the variables $p_1, \ldots, p_n$ and all the p_{ij}'s might occur in each of the equations $F_j(x_1, \ldots, x_n) = 0$, so they could not be associated with any of the players associated with those variables. Instead one might have to introduce m new players with whom to associate these equations. (For example, this was the role of Elizabeth in the circle example.) Now the equations associated with the players whose variables are $p_1, \ldots, p_n$ are free; since $n > m$ these can be used to fix the variables associated with the m new players at $\frac{1}{2}$ (as was done to the variable e in the circle example). □

The values of D and D' in Theorem 2 are obtained by setting $d_{ij} = d$ for all i and j. Theorem 1 follows since $\mathbb{R}$ is semialgebraically isomorphic to $(-1, 1)$ by the change of variables $t \mapsto t/(1-t^2)$ and $(-1, 1)$ is isomorphic to $(0, 1)$ by the change of variables $t \mapsto (t+1)/2$; then since the new x_i's take values in $(0, 1)$, their sum $\sum_{i=1}^{n} x_i$ takes values in some interval $(0, \delta)$, and dividing them all by δ lets us achieve the hypotheses of Theorems 6 and 7. Now the map $t \mapsto t/(1-t^2)$, when considered as a map from $\mathbb{R}$ to the whole line $\mathbb{R}$, is not one-to-one but one-to-two. So the image of our real algebraic variety under this map will have several pieces, but the piece lying in the interior of the n-cube $(-1, 1) \times \cdots \times (-1, 1)$ will be semialgebraically isomorphic to the original variety (and will not be connected to any other piece). Note that when the real algebraic variety is given by no more equations than unknowns, the isomorphism we exhibit in Theorems 6 and 7 is a homeomorphism.

In the game constructed in the proof of Theorem 7, the payoffs for many of the players depend only on the mixed strategies chosen by two or three of the other players. This is why the game is not generic. At the same time, this can happen very naturally in situations where the players interact locally. For example, a manufacturer making a particular product interacts with various suppliers who make the components that go into that product. These local interactions can be described by a graph. Such *graphical models* are studied by Kearns, Littman, and Singh [KLS01].

As mentioned before, in many cases games of smaller formats can be used. In particular, we restate Theorem 3 and prove it here:

Theorem 8. *Let S be the set of those roots of one polynomial equation $\alpha_d a^d + \cdots + \alpha_0 = 0$ in one unknown a which are real and lie in the interval $(0, 1)$. Then S is the set of first coordinates of the totally mixed Nash equilibria of a 3-person game in which Alice has two pure strategies and the Bob and Critter each have $\lceil d/2 \rceil + 1$ pure strategies.*

Proof. Suppose d is even, say $d = 2e$. We set Alice's payoffs so that equating them yields $c_e = b_1$. We set Bob's payoffs so that equating them yields the e equations

$$\begin{aligned} 0 &= \alpha_0 + a(s_1c_1 + \delta_1), \\ s_1c_1 + \delta_1 &= \alpha_1 + a(s_2c_2 + \delta_2), \\ &\vdots \\ s_{e-1}c_{e-1} + \delta_{e-1} &= \alpha_{e-1} + a(s_ec_e + \delta_e); \end{aligned}$$

and we set Critter's payoffs so that equating them yields the e equations

$$\begin{aligned} s_eb_1 + \delta_e &= \alpha_e + a(s_{e+1}b_2 + \delta_{e+1}), \\ s_{e+1}b_2 + \delta_{e+1} &= \alpha_{e+1} + a(s_{e+2}b_3 + \delta_{e+2}), \\ &\vdots \\ s_{2e-2}b_{e-1} + \delta_{2e-2} &= \alpha_{2e-2} + a(s_{2e-1}b_e + \delta_{2e-1}), \\ s_{2e-1}b_e + \delta_{2e-1} &= \alpha_{2e-1} + a\alpha_{2e}. \end{aligned}$$

As in the proof of Theorem 6, the s_i's and δ_i's are constants with $s_i \neq 0$ chosen so that $0 < b_i$, $0 < c_i$, $\sum_{i=1}^{e} b_i < 1$, and $\sum_{i=1}^{e} c_i < 1$, for all $a \in S$.

Suppose d is odd, say $d = 2e - 1$. Then we replace the last of Critter's equations by

$$b_e = \frac{1}{2} - \frac{1}{2}(b_1 + \cdots + b_{e-1}).$$

□

In the proof of Theorem 6, Alice's and Bob's equations were essentially wasted. On the other hand, in the proof of Theorem 8, we started the same way, multiplying out the polynomial according to Horner's rule and accumulating the result in one of the c_i's. But then we used one of Alice's equations to transfer the result to the b_i's, so we could continue the calculation using these variables. This can be done in many cases (although in general there may be more results that would have to be transferred than Alice's equations could accomodate). For instance, this is one reason why we were able to encode the circle in a 3-person game in which each player has three pure strategies; Theorem 6 would have predicted that Alice would need three pure strategies, Bob would need eight, and Critter would need nine. The other reason is that the polynomial $a_1\left(a_1 - \frac{1}{2}\right) + a_2\left(a_2 - \frac{1}{2}\right) + \frac{7}{64}$ can be written as the sum of a polynomial in a_1 and a polynomial in a_2. So we don't need to evaluate a polynomial of degree $d_2 = 2$ in a_2 for each of the $d_1 = 2$ powers of a_1; we just need to evaluate a polynomial in a_1 and a polynomial in a_2 separately.

Although we have stated our results in the geometric language of varieties, our proofs are purely algebraic and require few assumptions. Experts may note that this means our results are actually more general than what we have stated: they concern *schemes*, which generalize varieties.

2.5 Conclusion

Although the set of totally mixed Nash equilibria might comprise an arbitrary real algebraic variety, this does not mean it cannot be computed. As mentioned before, generically this set is finite. It is the set of solutions to a system of polynomial equations, which can be found, for instance, using polyhedral homotopy continuation software such as PHCpack by Verschelde [Ver99], as we discuss in Chapter 4 and in [Dat03b]. Sommese and Verschelde [SV00] have extended these methods to positive dimensional algebraic sets (e.g., curves and surfaces rather than isolated points). Indeed, how to solve systems of polynomial equations is a very active area of research.

Since Nash equilibria are usually not unique, the way that players approach equilibrium dynamically in repeated games under assumptions of imperfect information and/or bounded rationality has been studied both theoretically and experimentally. For example, Kalai and Lehrer [KL93] showed that under certain assumptions, "rational learning leads to Nash equilibrium." The existence of equilibrium sets with varying geometry and topology suggests that in these same dynamical models, interesting phenomena might continue to occur *after* equilibrium has been reached.

Chapter 3

Games and Graphs

The set of Nash equilibria for a game with generic payoff functions is finite [Har73]. This implies that the set of totally mixed Nash equilibria for a game with generic payoff functions is also finite. These are the real solutions to a system of polynomial equations and inequalities. The complex solutions to the system of equations are called *quasiequilibria*. Thus, the set of totally mixed Nash equilibria is a subset of the set of quasiequilibria. In fact, the set of quasiequilibria is also finite in the most generic case, and its cardinality can be computed as a function of the numbers of pure strategies of the players. Thus, this is an upper bound on the number of totally mixed Nash equilibria. Even in a nongeneric case, as long as the set of quasiequilibria is finite, its cardinality will be bounded above by the number in the generic case.

For the main theorem of this chapter, Theorem 9, we hypothesize a set of technical conditions that a system of polynomial equations may satisfy, which are encoded in an associated graph, the *polynomial graph*, and we prove a formula describing the number of solutions in this case. We then show how to associate such a graph to three special classes of games. The first two are graphical games and extensive-form games. The last is games with *emergent node tree structure*, a new model for games in which the players can be hierarchically decomposed into groups. Usually such hierarchical decomposition is modelled by *cooperative games*, and we briefly discuss how our model is related to, yet differs from, the cooperative framework.

3.1 Generic Number of Quasiequilibria

McKelvey and McLennan [MM97] have computed the exact number of quasiequilibria for games in the most generic case. The following theorem generalizes theirs to the situation in which the payoff matrices have more structure.

Theorem 9. *Suppose that* $0 < d \in \mathbb{N}$ *and that we are given a partition* $\{1, \ldots, d\} = \coprod_{i=1}^{N} T_i$ *of* $\{1, \ldots, d\}$. *Write* $d_i = |T_i|$. *Suppose further that we are given a directed graph G, the* polynomial graph, *on d vertices, denoted* $v_1, \ldots, v_d$, *without self-loops and with the property that for any* v_j *and* T_i, *if there is some* $k \in T_i$ *such that there is an edge from* v_j *to* v_k *in G, then for every* $k \in T_i$ *there is an edge from* v_j *to* v_k *in G. Let*

$$\begin{aligned} f_1(\sigma_1, \ldots, \sigma_d) &= 0, \\ f_2(\sigma_1, \ldots, \sigma_d) &= 0, \\ &\vdots \\ f_d(\sigma_1, \ldots, \sigma_d) &= 0 \end{aligned}$$

be a system (3.1) of d polynomial equations in d variables $\sigma_1, \ldots, \sigma_d$ *with the following properties:*

1. *All monomials occuring in the* f_i*'s are squarefree.*

2. *If* $\sigma_j, \sigma_k \in T_i$ *with* $j \neq k$ *then* σ_j *and* σ_k *do not both occur in any monomial of any of the* f_i*'s.*

3. *If there is no edge from* v_j *to* v_k *in G then the variable* σ_k *does not occur in* f_j.

Thus, the equations are multilinear, and they are linear over the variables from each T_i. *Construct a* $d \times d$ *matrix M as follows: If variable* σ_k *occurs in the polynomial* f_j, *with* T_i *the subset containing* v_k, *then*

$$M_{jk} = \frac{1}{(d_i!)^{1/d_i}},$$

otherwise $M_{jk} = 0$. *If the system (3.1) is* 0*-dimensional, then the number of its solutions in* $(\mathbb{C}^*)^d$ *(i.e. such that* $\sigma_k \neq 0$ *for all k) is bounded above by the permanent of M, and is equal to the permanent of M for generic coefficients.*

Proof. Without loss of generality, assume

$$T_i = \left\{ 1 + \sum_{l=1}^{i-1} d_l, 2 + \sum_{l=1}^{i-1} d_l, \ldots, d_i + \sum_{l=1}^{i-1} d_l \right\},$$

that is, that the T_i's are contiguous.

Let $a_{ij} = 1$ if there is an edge in G from v_j to v_k for $k \in T_i$, and $a_{ij} = 0$ otherwise. Then the Newton polytope P_j of f_j is the Cartesian product $P_{1j} \times P_{2j} \times \cdots \times P_{Nj}$, where P_{ij} is the convex hull of the scaled coordinate vectors $\{a_{ij} e_k \mid k \in T_i\}$ and the origin. For i with $a_{ij} = 1$, P_{ij} is the d_i-dimensional unit simplex, and for i with $a_{ij} = 0$, P_{ij} degenerates to the d_i-dimensional origin (which is a 0-dimensional simplex). By the Bernstein-Kouchnirenko Theorem [Ber75] [Kou76], it suffices to show that the mixed volume of the polytopes $P_1, \ldots, P_d$ is given by the permanent of M.

Let $Q_j = \lambda_1 P_1 + \cdots + \lambda_j P_j$, where $+$ denotes Minkowski addition and the scale factors $\lambda_1, \ldots, \lambda_j$ are parameters. We show by induction on j that $Q_j = Q_{1j} \times Q_{2j} \times \cdots \times Q_{Nj}$, where Q_{ij} is the convex hull of

$$\{(a_{i1}\lambda_1 + a_{i2}\lambda_2 + \cdots + a_{ij}\lambda_j)e_k \mid k \in T_i\}$$

and the origin. (If $a_{i1}\lambda_1 + a_{i2}\lambda_2 + \cdots + a_{ij}\lambda_j = 0$ then Q_{ij} degenerates to the origin.) The base case follows from our characterization of P_j above. Now consider the Minkowski sum of $Q_j = Q_{1j} \times \cdots \times Q_{Nj}$ and $\lambda_{j+1} P_{j+1} = (\lambda_{j+1} P_{1(j+1)}) \times \cdots \times (\lambda_{j+1} P_{N(j+1)})$. It follows from the definition of Minkowski sum that this is $(Q_{1j} + \lambda_{j+1} P_{1(j+1)}) \times \cdots \times (Q_{Nj} + \lambda_{j+1} P_{N(j+1)})$, and (using the induction hypothesis) that each factor $Q_{ij} + \lambda_{j+1} P_{i(j+1)}$ is equal to the convex hull of

$$\{(a_{i1}\lambda_1 + a_{i2}\lambda_2 + \cdots + a_{ij}\lambda_j + a_{i(j+1)} e_{j+1}) e_k \mid k \in T_i\}$$

and the origin.

The d_i-dimensional volume of the d_i-dimensional unit simplex scaled by λ in each dimension is

$$\frac{\lambda^{d_i}}{(d_i)!}.$$

We are interested in the d-dimensional volume of Q_d. If $a_{i1} = a_{i2} = \cdots = a_{id} = 0$ for some i, then this volume vanishes, and hence the mixed volume also vanishes. In this case the kth column of the matrix M will be all zeroes for any $k \in T_i$, so the permanent of M also vanishes, and the theorem holds. So assume that for each i, there is some j with $a_{ij} = 1$. Then the volume of Q_d is

$$\prod_{i=1}^{N} \frac{(a_{i1}\lambda_1 + \cdots + a_{id}\lambda_d)^{d_i}}{d_i!}.$$

Let (g_{jk}) be the adjacency matrix of G, that is, $g_{jk} = 1$ if there is an edge in G from v_j to v_k and $g_{jk} = 0$ otherwise. Then $a_{ij} = g_{jk}$ for all $k \in T_i$. So the volume of Q_d is

$$\frac{\prod_{k=1}^{d} \left(g_{1k}\lambda_1 + \cdots + g_{dk}\lambda_d\right)}{\prod_{i=1}^{N} d_i!}.$$

The mixed volume of $P_1, \ldots, P_d$ is the coefficient of $\lambda_1\lambda_2\cdots\lambda_d$ in the above expression, which is the permanent of (g_{jk}) divided by $\prod_{i=1}^{N} d_i!$.

It remains to show that the permanent of M is the permanent of (g_{jk}) divided by $\prod_{i=1}^{N} d_i!$. Note that $M_{jk} \neq 0$ exactly when $g_{jk} \neq 0$. We induct on N. For the base case, $d_1 = d$, and each nonzero entry of M is $(1/d!)^{1/d}$. A term in the permanent of M is the product of d entries from M, so if it is nonzero it is $1/d!$. Thus the permanent of M is $1/d!$ times the permanent of (g_{jk}), as required. Now partition the matrix M and the matrix (g_{jk}) into two vertical bands corresponding to the subsets $\cup_{i=1}^{N-1} T_i$ and T_N. The permanent can be computed as the sum of a term for each choice of d_N rows $1 \leq j_1 < \cdots < j_{d_N} \leq d$: compute the $(d - d_N) \times (d - d_N)$ subpermanent of the left band obtained by crossing out those rows, compute the $d_N \times d_N$ subpermanent of the right band corresponding to those rows, and multiply them together. By the inductive hypothesis, the left subpermanent of M is the left subpermanent of (g_{jk}) divided by $\prod_{i=1}^{N-1} d_i!$. For the right subpermanent, every row is either all nonzero or all zero. If any row is all zero, both right subpermanents vanish. If every entry is nonzero, then all the entries are the same: $g_{jk} = 1$ and $M_{jk} = (1/d_N!)^{1/d_N}$. The right subpermanent of M is $d_N! \left((1/d_N!)^{1/d_N}\right)^{d_N} = 1$, and the right subpermanent of (g_{jk}) is $d_N!$. So the whole term for M is the whole term for (g_{jk}) divided by $\prod_{i=1}^{N} d_i!$. □

We note that if the coefficients are generic subject to the conditions given in Theorem 9, *all* the solutions to the system will lie in the torus $(\mathbb{C}^*)^d$. In what follows we will refer to "the number of solutions in the torus $(\mathbb{C}^*)^d$" as "the number of solutions" by abuse of language.

Corollary 10. *Convert the directed graph G of Theorem 9 into a bipartite graph on $2d$ vertices, with the source of every edge on the left side and the target of every edge on the right side. If the system in Theorem 9 is 0-dimensional with generic coefficients, then it has a solution if and only if this bipartite graph has a perfect matching.*

Proof. From the proof of Theorem 9, we see that the number of solutions is nonzero if and only if the permanent of the adjacency matrix is nonzero. It is a well-known fact that this is equivalent to the existence of a perfect matching: any permutation π which contributes a nonvanishing term $\prod_{j=1}^{d} g_{j\pi(j)}$ to the permanent corresponds to a perfect matching, where vertex j on the left is matched to vertex $\pi(j)$ on the right. □

In fact, we could have used the bipartite graph in Theorem 9. However, we defined the polynomial graph to be the directed graph to remain consistent with the usual definition of graphical models of games.

Corollary 11. *If the system in Theorem 9 is 0-dimensional and has a solution, then every node in the graph G lies on a directed cycle.*

Proof. As in the proof of the previous corollary, a permutation π must exist such that j has an edge to $\pi(j)$ for every $j = 1, \ldots, d$. This permutation can be expressed as a product of disjoint cycles. Each node lies in one of these cycles, and a cycle of the permutation corresponds to a directed cycle in the graph. □

We should note carefully that the Bernstein-Kouchnirenko theorem gives the number of solutions to a *0-dimensional* polynomial system. So when the number given by that theorem—in particular, the permanent of the matrix in Theorem 9—vanishes, *either* the polynomial system has no solution, *or* its solution set has positive dimension.

Note that the conditions on G imply that the matrix M has a $d_i \times d_i$ block of zeroes along its diagonal for $i = 1, \ldots, N$. This is because G has no self-loops, and if it had an edge from an element v_j of T_i to any other element v_k of T_i, then there would have to be an edge from v_j to every element of T_i including itself.

Our theorem applies to games. In this case, each T_i corresponds to the set of strategies of player i. The blocks of zeroes along the diagonal imply that a player's expected payoffs from their own pure strategies do not depend on the probabilities they have assigned to their own pure strategies, so these polynomial systems do indeed correspond to the equations for totally mixed Nash equilibria of games.

Corollary 12. *Consider a normal form game between players $I = \{1, \ldots, N\}$ with pure strategy sets S_i for each i and generic utility functions $u_i\colon \prod_{i \in I} S_i \to \mathbb{R}$. Construct a graph G with nodes $\coprod_{i \in I}(S_i - \{s_{i0}\})$ such that there is an edge from s_{ik} to s_{jl} in G if and only if $i \neq j$. Let the variable corresponding to s_{ik} be $\sigma_i(s_{ik})$ and the equation corresponding to s_{ik} be the indifference equation $u_i(s_{ik}, \sigma_{-i}) = u_i(s_{i0}, \sigma_{-i})$. Then this system of equations obeys the conditions of Theorem 9, so the number of solutions in the generic case is given by that theorem.*

This special case was proved as Theorem 2 in [McL99], so our Theorem 9 is a generalization of that theorem.

3.2 Graphical Games

Kearns, Littman, and Singh [KLS01] defined the concept of *graphical games*, or games obeying *graphical models*. (That paper considers undirected graphs, but the extension to directed graphs which we will use is straightforward.) A game between players $1, \ldots, N$ obeys a directed graphical model, if the payoffs to player i_1 only depend on the actions of those players $i_2 \neq i_1$ for which there is an edge from i_1 to i_2 in the graphical model.

Our theorem applies in particular to graphical games. As in Corollary 12, we take the pure strategy sets S_i to be the sets T_i of Theorem 9. Given a polynomial graph G as in Theorem 9, we draw an edge from i_1 to i_2 in the graphical model if there is any $j \in T_{i_1}$ with edges to the vertices in T_{i_2} in the polynomial graph G. The polynomial graph G may not represent the most generic case of the graphical model, however. If we are given a graphical model, then to construct its polynomial graph G, for any edge from i_1 to i_2, we draw edges in G from *every* vertex $j \in T_{i_1}$ to every vertex in T_{i_2}.

Corollary 13. *Suppose a normal form game between players $I = 1, \ldots, N$ with pure strategy sets S_i for each i and utility functions $u_i \colon \prod_{i \in I} S_i \to \mathbb{R}$ obeys a directed graphical model γ with nodes $1, \ldots, N$. Construct a graph G with nodes $\coprod_{i \in I} S_i$ such that there is an edge from s_{ik} to s_{jl} in G if and only if there is an edge from i to j in γ. Then the system of equations defining the quasiequilibria of G satisfies the hypotheses of Theorem 9, so the number of such quasiequilibria in the generic case is given by the permanental formula.*

For example, consider a game with 4 players, each with 3 pure strategies. Generically, such a game has

$$\operatorname{per}\begin{pmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \end{pmatrix} = 297$$

quasiequilibria.

But suppose now that game obeys a graphical model as in Figure 3.1. The nodes in the graphical model refer to the players, and the edges specify that the payoff to the source player depends on the actions of the target player. For brevity, write $a = \sigma_1(s_{11})$, $b = \sigma_2(s_{12})$, $c = \sigma_2(s_{21})$, $d = \sigma_2(s_{22})$, $e = \sigma_3(s_{31})$, $f = \sigma_3(s_{32})$, $g = \sigma_4(s_{41})$, and $h = \sigma_4(s_{42})$. Since the payoff to player 1 depends only on the actions of player 2, equating the payoff to player 1 from pure strategies s_{10} and s_{11} gives

$$\begin{aligned} & u_1(s_{10}, s_{20}, \bullet)\, \sigma_2(s_{20}) + u_1(s_{10}, s_{21}, \bullet)\, \sigma_2(s_{21}) + u_1(s_{10}, s_{22}, \bullet)\, \sigma_2(s_{22}) \\ = \quad & u_1(s_{11}, s_{20}, \bullet)\, \sigma_2(s_{20}) + u_1(s_{11}, s_{21}, \bullet)\, \sigma_2(s_{21}) + u_1(s_{11}, s_{22}, \bullet)\, \sigma_2(s_{22}) \end{aligned}$$

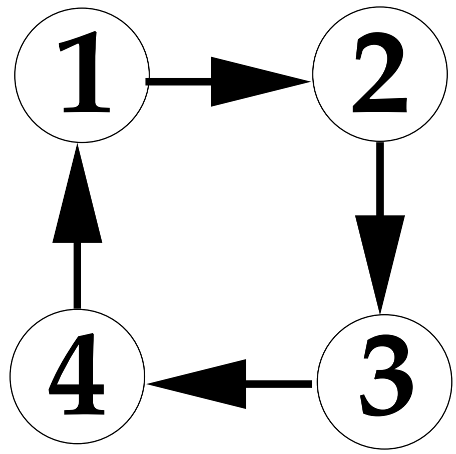

Figure 3.1: Graphical game

or

$$(u_1(s_{11}, s_{20}, \bullet) - u_1(s_{10}, s_{20}, \bullet))\,(1 - c - d) + \\ + (u_1(s_{11}, s_{21}, \bullet) - u_1(s_{10}, s_{21}, \bullet))\,c + (u_1(s_{11}, s_{22}, \bullet) - u_1(s_{10}, s_{22}, \bullet))\,d = 0.$$

Thus for player 1 we have two equations of the form

$$\bullet c + \bullet d + \bullet = 0,$$

for player 2 we have two equations of the form

$$\bullet e + \bullet f + \bullet = 0,$$

for player 3 we have two equations of the form

$$\bullet g + \bullet h + \bullet = 0,$$

and for player 4 we have two equations of the form

$$\bullet a + \bullet b + \bullet = 0.$$

Then the associated polynomial graph is depicted in Figure 3.2. The equation asso-

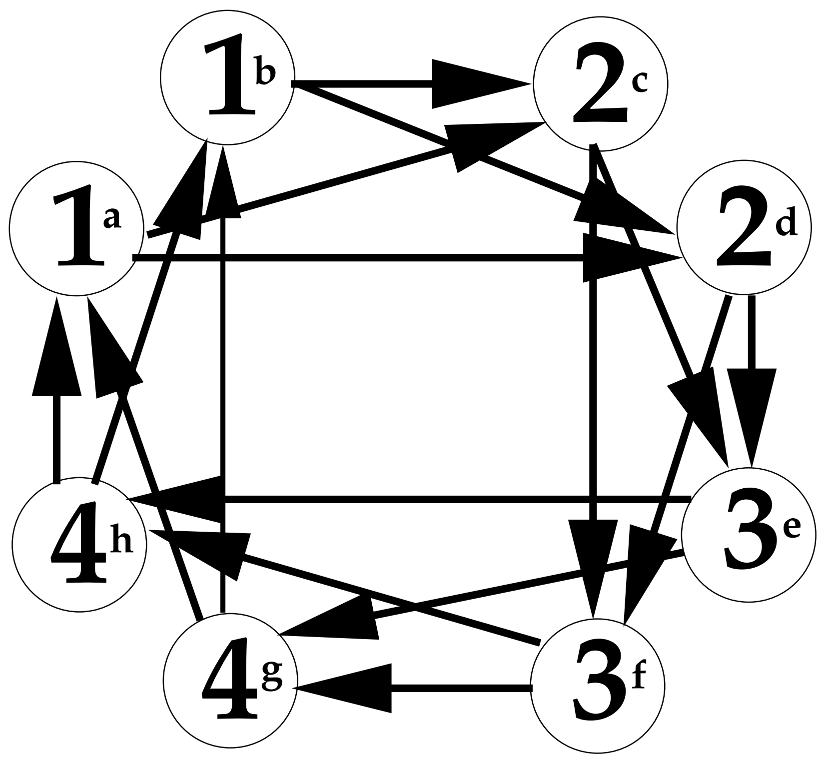

Figure 3.2: Associated polynomial graph for graphical game

ciated with the node labelled $1a$ equates the payoffs to player 1 from choosing s_{11} (which 1 does with probability a) or choosing s_{10}. The game has

$$\operatorname{per}\begin{pmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = 1$$

quasiequilibrium. Indeed, this will always be the case for a graphical model which is a directed cycle, where each player has the same number of pure strategies. The reason is that the indifference equations in this case are *linear*, as we saw in this example.

The polynomial graph G as defined in Theorem 9 contains more refined information than the graphical model. The partition into the T_i's also can be more refined than the partition of the set of all pure strategies into the sets of pure strategies for each player. Next we will see an example of such a refinement when considering the reduction of extensive-form games to normal-form, where actions correspond to branches of the game tree.

3.3 Extensive-Form Games

Now we consider extensive-form games. We begin by noting the following:

Theorem 14. *All totally mixed Nash equilibria of an extensive form game are subgame perfect.*

Proof. Let σ be a totally mixed Nash equilibrium of an extensive form game with N players defined by game tree T. Note that the strategy profile induced by σ on every subgame is also totally mixed. Let ν be a non-leaf node of T. Let $\tilde{\sigma}$ be the strategy profile induced by σ in the subgame induced by ν. Let $\tilde{s}_j$ and $\tilde{t}_j$ be pure strategies of player j in this subgame. Choose an action for j at each node μ that is not a descendant of ν where j acts, such that if μ is an ancestor of ν then j chooses the branch leading towards ν, and use this choice to extend $\tilde{s}_j$ and $\tilde{t}_j$ to pure strategies s_j and t_j of player j in the original game. (So, s_j and t_j specify the same actions outside the subtree.) Let $\nu_0 \dots \nu_m = \nu$ be the unique path from the root ν_0 of T to ν. We have $u_j(s_j, \sigma_{-j}) = u_j(t_j, \sigma_{-j})$. Let L be the set of all leaves of T under ν and L' be the set of all other leaves. Then

$$\begin{aligned} u_j(s_j, \sigma_{-j}) &= \textstyle\sum_{\lambda \in L} u_j(\lambda) \Pr[\lambda|(s_j, \sigma_{-j})] + \sum_{\lambda \in L'} u_j(\lambda) \Pr[\lambda|(s_j, \sigma_{-j})] \\ &= \textstyle\sum_{\lambda \in L} u_j(\lambda) \Pr[\lambda|(s_j, \sigma_{-j})] + \sum_{\lambda \in L'} u_j(\lambda) \Pr[\lambda|(t_j, \sigma_{-j})] \end{aligned}$$

since s_j and t_j choose the same actions outside the subtree. Thus

$$\sum_{\lambda \in L} u_j(\lambda) \Pr[\lambda|(s_j, \sigma_{-j})] = \sum_{\lambda \in L} u_j(\lambda) \Pr[\lambda|(t_j, \sigma_{-j})]. \tag{3.1}$$

Furthermore, for any $\lambda \in L$, we have

$$\begin{aligned} \Pr[\lambda|(s_j, \sigma_{-j})] &= \Pr[\lambda|(\tilde{s}_j, \tilde{\sigma}_{-j})] \prod_{k=0}^{m-1} \Pr[\nu_k \to \nu_{k+1}|(s_j, \sigma_{-j})] \\ &= \Pr[\lambda|(\tilde{s}_j, \tilde{\sigma}_{-j})] \prod_{k=0}^{m-1} \Pr[\nu_k \to \nu_{k+1}|(t_j, \sigma_{-j})]. \end{aligned}$$

Noting that the common factor $\prod_{k=0}^{m-1} \Pr[\nu_k \to \nu_{k+1}|(t_j, \sigma_{-j})]$ in equation (3.1) is positive by our choice of s_j, t_j and because σ is totally mixed, we have that

$$\begin{aligned} u_j(\tilde{s}_j, \tilde{\sigma}_{-j}) &= \sum_{\lambda \in L} u_j(\lambda) Pr[\lambda|(\tilde{s}_j, \tilde{\sigma}_{-j})] \\ &= \sum_{\lambda \in L} u_j(\lambda) Pr[\lambda|(\tilde{t}_j, \tilde{\sigma}_{-j})] \\ &= u_j(\tilde{t}_j, \tilde{\sigma}_{-j}). \end{aligned}$$

Thus $\tilde{\sigma}$ is a (totally mixed) Nash equilibrium of the subgame induced by ν. □

In light of this observation, the divide-and-conquer approach to finding all Nash equilibria of a normal form game can be modified in the spirit of backwards induction to finding all subgame perfect equilibria (including mixed ones) of an extensive form game. Recall that in a normal form game, we would consider subproblems in which one pure strategy of one player i was removed. Now we instead consider subproblems in which, for some edge $\nu \to \mu$ where i acts at ν, we delete that edge and the entire subtree below μ. We compute the normal form for the game described by this pruned tree and recursively find all its subgame perfect equilibria. Each such equilibrium σ induces an equilibrium $\tilde{\sigma}$ in the subgame under ν in the pruned tree. To check whether σ is an equilibrium of the original game, we recursively compute all the equilibria of the subgame under μ (where i does not act), and check that for each such equilibrium τ, we have $u_i(\tilde{\sigma}) \geq u_i(\tau)$.

We saw during the above proof that for a totally mixed strategy profile σ, the equations $u_j(s_j, \sigma_{-j}) = u_j(t_j, \sigma_{-j})$ for all pure strategies s_j, t_j of j imply the corresponding equations for each subtree. The converse implication also clearly holds.

We will now associate a polynomial graph to a system of equations for the quasiequilibria of an extensive-form game, so that we can apply Theorem 9. For each node in the game tree where a player acts, we will have a variable for every edge emanating from that node except one distinguished edge. This is because the sum of the probabilities of choosing each of those edges must be 1, so we eliminate one variable. Thus, we compare the payoffs between choosing the distinguished edge and choosing any other edge. The equations will be indifference equations for subgames of the extensive-form game.

Theorem 15. *The set of quasiequilibria of a generic extensive-form game is either empty or has positive dimension.*

Proof. Consider an extensive form game with players $I = 1, \ldots, N$ and game tree T. Let A be the set of non-leaf nodes in T not labelled by 0. For each $\nu \in A$, let $E(\nu)$ be the set of edges emanating from ν. For each $\nu \in A$, let i be the player which acts at ν and pick an element $e_{i\nu} \in E(\nu)$. Let $d = \sum_{\nu \in A} |E(\nu) - 1|$ and partition d as $\coprod_{\nu \in A} (E(\nu) - \{e_{i\nu}\})$. Define a directed graph G on a set of d vertices

$$\bigcup_{\nu \in A} \{n_e \mid e \in E(\nu) - \{e_{i\nu}\}\}$$

as follows: there is an edge from n_e with $e \in E(\nu) - \{e_{i\nu}\}$ to $n_{e'}$ with $e' \in E(\mu) - \{e_{j\mu}\}$ if $i \neq j$, ν is an ancestor of μ, either e or $e_{i\nu}$ lies on the path from ν to μ, and if i

acts at some node κ between ν and μ, then the edge $e_{i\kappa}$ lies on the path from ν to μ. We will define a system of equations equivalent to the equations defining totally mixed Nash equilibria of the extensive form game and satisfying conditions 1 to 3 of Theorem 9. The polynomial graph G is acyclic, so Corollary 11 implies our assertion.

First we must state what the equations are. Fix a node $\nu \in A$ and let i be the player which acts at ν. Then $|E(\nu)| - 1$ equations refer to the subgame induced by this node. For each $e \in E(\nu)$, define the pure strategy s_{ie} of i in this subgame by $s_{ie}(\nu) = e$ and $s_{ie}(\mu) = e_{i\mu}$ for any node μ below ν where i acts. Writing $\tilde{\sigma}$ for the strategy profile induced by σ in the subgame under ν, the $|E(\nu)| - 1$ equations are the equations $u_i(s_e, \tilde{\sigma}_{-i}) = u_i(s_{e_{i\nu}}, \tilde{\sigma}_{-i})$ for $e \in E(\nu) - \{e_{i\nu}\}$. In these equations we eliminated $\sigma(e_{j\mu})$ for every μ below ν where i does not act, by substituting $1 - \sum_{e \in \tilde{E}(\mu) - e_{\{j\mu\}}} \sigma_j(e)$ for $\sigma_j(e_{j\mu})$.

These are some of the indifference equations for the subtree below ν, which as we saw in the previous theorem are implied by the indifference equations for the whole tree. We show by induction that these equations also imply all the indifference equations for the subtree below ν. (Thus we will have the indifference equations for every subtree, and hence the whole tree, i.e., the original game.) Firstly, i is indifferent between *all* i's pure strategies in the subgame below ν, because although we fixed i's pure strategies at nodes μ below ν where i acts to be $e_{i\mu}$, we also have that i is indifferent between i's pure strategies in the subgame below μ by the induction hypothesis. Secondly, consider any other player j. Let $\mu_1, \ldots, \mu_m$ be the nodes below ν where j acts, such that j does not act at any node between ν and μ_k for any k. Let $\tilde{s}_j, \tilde{t}_j$ be pure strategies of j in the subgame below ν, and write $\tilde{s}_{j_k}, \tilde{t}_{j_k}$ for the respective induced pure strategies of j in the subgame below μ_k. So $\tilde{s}_j = (\tilde{s}_{j_1}, \ldots, \tilde{s}_{j_m})$ and $\tilde{t}_j = (\tilde{t}_{j_1}, \ldots, \tilde{t}_{j_m})$. Write the set L of leaves below ν as $L = L_0 \cup \bigcup_{k=1}^{m} L_k$, where L_0 is the set of leaves λ such that j does not act between ν and λ and L_k is the set of leaves below μ_k for $k = 1, \ldots, m$. Then

$$\begin{aligned} u_j(\tilde{s}_j, \tilde{\sigma}_{-j}) &= \sum_{\lambda \in L} u_j(\lambda | \tilde{s}_j, \tilde{\sigma}_{-j}) \\ &= \sum_{\lambda \in L_0} u_j(\lambda | \tilde{\sigma}_{-j}) + \sum_{k=1}^{m} \sum_{\lambda \in L_k} u_j(\lambda | \tilde{s}_{j_k}, \tilde{\sigma}_{-j}) \\ &= \sum_{\lambda \in L_0} u_j(\lambda | \tilde{\sigma}_{-j}) + \sum_{k=1}^{m} \sum_{\lambda \in L_k} u_j(\lambda | \tilde{t}_{j_k}, \tilde{\sigma}_{-j}) \\ &= u_j(\tilde{t}_j, \tilde{\sigma}_{-j}) \end{aligned}$$

since for each k, $\sum_{\lambda \in L_k} u_j(\lambda | \tilde{s}_{j_k}, \tilde{\sigma}_{-j}) = \sum_{\lambda \in L_k} u_j(\lambda | \tilde{t}_{j_k}, \tilde{\sigma}_{-j})$ by the induction hypothesis.

We can already see that the set of solutions to these equations, if nonempty, is positive-dimensional. If player i acts at the root ν, then for any edge e emerging from ν, $\sigma_i(e)$ does not appear in any of the equations.

All the monomials occurring in these equations are squarefree. For each leaf λ under ν, let the path from ν to λ be $\nu = \nu_1 \ldots \nu_k = \lambda$. Then for any player j with pure strategy $\tilde{s}_j$, we have $\Pr[\lambda | \tilde{s}_j, \tilde{\sigma}_{-j}] = \prod_{l=1}^{k-1} \Pr[\nu_l \to \nu_{l+1} | \tilde{s}_j, \tilde{\sigma}_{-j}]$, and each nonconstant term in the product is $\sigma_n(\nu_l \to \nu_{l+1})$ for some player $n \neq j$. So for any edge e where n acts, the variable $\sigma_n(e)$ occurs at most once in such a product. In fact $\sigma_n(e)$ occurs in such a product for at most one $e \in E(\nu_l)$. (That is, if $e, e' \in E(\nu_l)$ then $\sigma_n(e)$ and $\sigma_n(e')$ do not both occur in this monomial. So condition 2 of Theorem 9 holds.) When we eliminate $\sigma_n(e_{n\nu_l})$, we replace it by an affine expression, so this remains true. Thus condition 1 of Theorem 9 holds.

The equations corresponding to $E(\nu) - \{e_{i\nu}\}$ concern only the subgame below ν, so $\sigma_j(\mu \to \kappa)$ occurs in these equations only if ν is an ancestor of μ. Furthermore, if i acts at κ below ν, then $\sigma_i(e)$ does not occur for any edge $e \in E(\kappa) - \{e_{i\kappa}\}$, since we fix that i chooses $e_{i\kappa}$. For the same reason $\sigma_j(e)$ does not occur for $e \in E(\mu) - \{e_{j\mu}\}$ for any μ that lies below κ but not below $e_{i\kappa}$. Thus condition 3 holds. □

Our result does not contradict Harsanyi's generic finiteness theorem [Har73], because generically, iterated elimination of weakly dominated strategies/backward induction will lead to a unique subgame perfect equilibrium (and so indeed there will be no totally mixed Nash equilibria). On the other hand, another way to look at our result is that in every *interesting* extensive-form game—one which is not completely solved by backward induction, giving a unique equilibrium—the set of totally mixed Nash equilibria is also interesting; it has positive dimension.

In particular, if ν is a node all of whose children are leaves, the equations corresponding to ν will be equations between constants, stating that for the player i who acts at ν, the utilities $u_i(\lambda)$ at all the leaves λ below ν must be equal. This is true if iterated elimination of strictly dominated pure strategies has already been performed on this game.

It is clear that the system of equations we obtained is not canonical, since we have made arbitrary choices of the edges $e_{i\nu}$ and the subtrees below each possible choice are different. Choosing a different system may make it easier to compute the set of quasiequilibria.

We now present an example where the set of totally mixed Nash equilibria is a positive-dimensional semialgebraic variety. Consider the extensive form game specified in Figure 3.3. The polynomial graph associated with this game tree is depicted in Figure 3.4. For brevity, we write for example $\sigma_1(C)$ for $\sigma_1(A \to C)$. The quasiequilibria obey a system of 4 equations as in Theorem 15. The equation associated with the edge $E \to G$ equates the payoff to player 3 from choosing this edge with that from choosing the edge $E \to F$, i.e., $u_3(F) = u_3(G)$. No variables occur in this equation, that is, it is an equation between constants. Similarly, the equation associated with the edge $E \to H$ is $u_3(F) = u_3(H)$. The equation associated with the edge $C \to E$ is $u_2(D) = u_2(E)$, where we have written $u_2(E)$ for the expected payoff $u_2(E, \sigma_{-2})$ to player 2 for choosing the edge $C \to E$, given the strategy profile of the other players. In this case $u_2(E) = u_2(F)\,\sigma_3(F) + u_2(G)\,\sigma_3(G) + u_2(H)\,\sigma_3(H)$, so

$$u_2(D) = u_2(F) + (u_2(G) - u_2(F))\ \sigma_3(G) + (u_2(H) - u_2(F))\ \sigma_3(H).$$

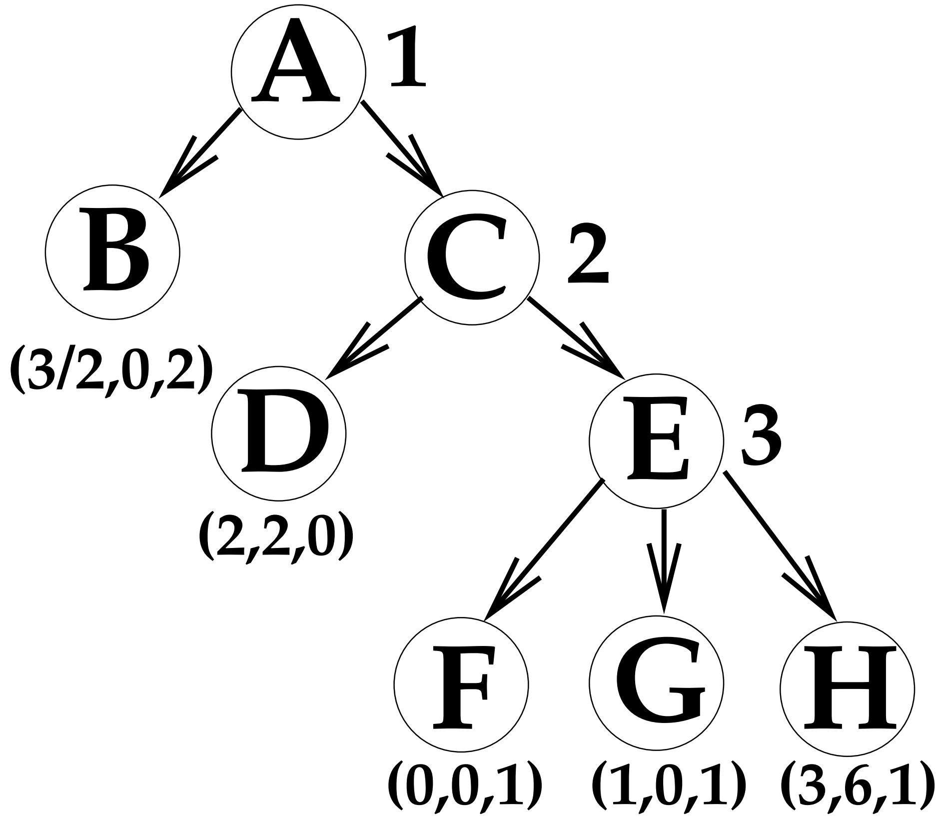

Figure 3.3: An Extensive Form Game

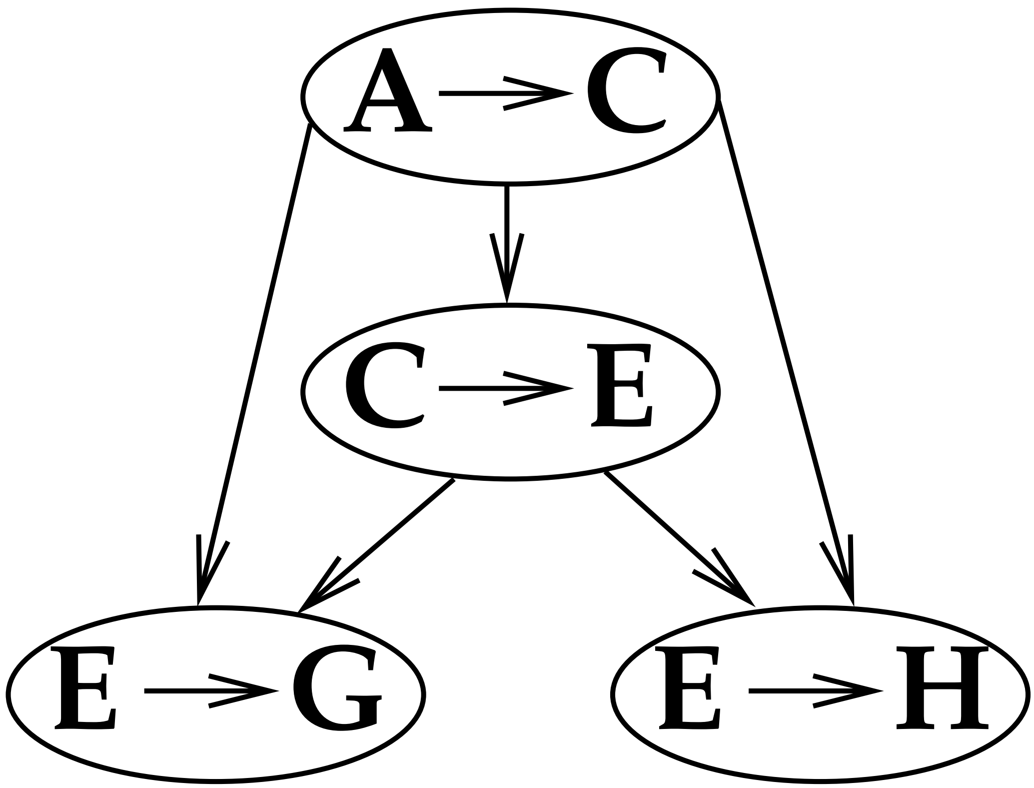

Figure 3.4: Associated Polynomial Graph For An Extensive Form Game

Finally, the equation associated to the edge $A \to C$ is

$$\begin{aligned} u_1(B) &= u_1(C) \\ &= u_1(D)\,(1 - \sigma_2(E)) + u_1(F)\,\sigma_2(E)\,(1 - \sigma_3(G) - \sigma_3(H)) \\ &\quad + u_1(G)\,\sigma_2(E)\,\sigma_3(G) + u_1(H)\,\sigma_2(E)\,\sigma_3(H). \end{aligned}$$

Looking at the specific payoffs in Figure 3.3, we see that the payoffs to player 3 for choosing F, G, or H are equal, as required. Equating the payoffs to player 2 for choosing D or E, we get $6\,\sigma_3(H) = 2$, or $\sigma_3(H) = \frac{1}{3}$. This leaves $\sigma_3(G)$ free to vary such that $0 < \sigma_3(G) < \frac{2}{3}$. Finally, we must equate the payoffs to player 1 for choosing B or C. This gives

$$2(1 - \sigma_2(E)) + \sigma_2(E)\,(\,\sigma_3(G) + 1) = \frac{3}{2}$$

or

$$\sigma_2(E)(1 - \sigma_3(G)) = \frac{1}{2}.$$

Thus the points $\sigma_3(G)$ and $\sigma_2(E)$ lie on a hyperbola. This hyperbola intersects the interior of the product of simplices. For instance, the point $\sigma_3(G) = \frac{5}{12}$ (so $\sigma_3(F) = \frac{1}{4}$) and $\sigma_2(E) = \frac{6}{7}$ lies in this intersection. So the set of quasiequilibria is a portion of a hyperbolic cylinder, the product of a segment of a hyperbola with a line segment (since $\sigma_1(B)$ varies freely with $0 < \sigma_1(B) < 1$).

We can analyze this game a little further. Player 3 would like player 1 to sometimes choose B, but cannot force player 1 always to choose B, since if player 2 always chooses D then both player 1 and player 2 are better off with player 1 choosing C. The best player 3 can do is make the payoffs to player 1 from choosing B and C equal. Now if player 3 made player 2 get a greater payoff from choosing D than E, then player 2 would always choose D, player 1 would always choose C, and player 3 would get nothing. So player 3 must make $u_2(D) \le u_2(E)$. We analyzed the case $u_2(D) = u_2(E)$ above. If player 3 makes $\sigma_3(H) > \frac{1}{3}$, then $u_2(D) < u_2(E)$ and player 2 will always choose E. Then the payoff to player 1 from choosing C is $\sigma_3(G) + 3\,\sigma_3(H)$. Thus we have $\sigma_3(G) + 3\,\sigma_3(H) = \frac{3}{2}$ with $\frac{1}{3} < \sigma_3(H) \le \frac{1}{2}$ (this makes $0 \le \sigma_3(G) < \frac{1}{2}$ and $\frac{1}{6} < \sigma_3(F) \le \frac{1}{2}$). Then $\sigma_1(C)$ varies freely with $0 \le \sigma_1(C) \le 1$, so we have a rectangle of partially mixed equilibria. Player 3 is better off choosing these, since then the outcome D where player 3 gets zero payoff is never reached. Along the line $\sigma_3(G) + 3\,\sigma_3(H) = \frac{3}{2}$, equilibria with greater $\sigma_3(H)$ *Pareto dominate* those with smaller $\sigma_3(H)$, i.e., they make some player better off and no player worse off. Specifically, the payoff to player 2 increases, the payoff to player 1 is always $\frac{3}{2}$, and the payoff to player 3 stays the same at $2(1 - \sigma_1(C)) + \sigma_1(C) = 2 - \sigma_1(C)$. Thus the Pareto dominant equilibrium among those on this line is that player 3 has $\sigma_3(F) = \frac{1}{2}$, $\sigma_3(G) = 0$, and $\sigma_3(H) = \frac{1}{2}$. On the other hand, at the pure strategy equilibrium where player 3 always chooses H, we have that player 1 always chooses C, and the payoff to player 3 falls from $2 - \sigma_1(C)$ to 1. Thus player 3 does not prefer this equilibrium, and instead mixes F and H equally to have some chance of a higher payoff. As $\sigma_1(C)$ increases, the payoff to player 3 decreases and the payoff to player 2 increases, so the equilibria along this line do not Pareto dominate each other. Thus without introducing other issues (such as risk-aversion) there is no criterion for predicting which of the equilibria along the line $0 < \sigma_1(C) < 1$, $\sigma_2(E) = 1$, $\sigma_3(F) = \sigma_3(H) = \frac{1}{2}$ should be chosen.

3.4 Games With Emergent Node Tree Structure

So far we have been discussing normal form games with finite numbers of players, each with a finite number of pure strategies. Such a game is defined by giving a set of players $I = \{1, \ldots, \mathbb{N}\}$, for each player i a finite set of pure strategies S_i, and for each pure strategy profile σ (element of the product $S = \prod_{i \in I} S_i$) and each player i the utility $u_i(\sigma)$ received by that player when that strategy profile is played. Now we will introduce a particular kind of structure that a normal form game may have.

We now define an *emergent node tree structure* on a normal form game. This is a new model for games in which the players can be hierarchically decomposed into groups. Usually such hierarchical decomposition is discussed in the framework of cooperative game theory. Instead, we define certain conditions on the payoff functions in a noncooperative game such that a given hierarchical decomposition “makes sense”, in a way that we will define precisely. At the end of this section we briefly describe how our framework relates to that of cooperative game theory.

Definition. An *emergent node tree structure* on a normal form game with player $I = \{1, \ldots, \mathbb{N}\}$, pure strategy sets S_i for $i \in I$, and utility functions $u_i \colon \prod_{i \in I} S_i \to \mathbb{R}$ to consist of:

- A tree T with N leaves. The leaves are in bijection with the players $I = \{1, \ldots, \mathbb{N}\}$. Write C_v for the set of children of a node $v \in T$, B_v for the set of its siblings, and $f(v)$ for its parent.

- For each non-leaf, non-root node v of the tree (which we call an *emergent player*), a set S_v of pure strategies, with $|S_v| \leq \prod_{w \in C_v} |S_w|$.

- For each non-leaf, non-root node v, for each element s_{C_v} of the product $S_{C_v} = \prod_{w \in C_v} S_w$ of the pure strategies of its children and each element s_{vk} of S_v, a number $p_v(k, s_{C_v})$ signifying the probability that the (emergent) strategy of the emergent player v is s_{vk} when the strategies of its children are given by s_{C_v}. So if v has K pure strategies, then $\sum_{k=1}^{K} p_v(k, s_{C_v}) = 1$. If the children of v execute a mixed strategy, then the emergent mixed strategy of v is given by multilinearity. Thus we have defined a linear map from the strategy space of the children to the strategy space of the parent. We require that this map have full rank.

- For each non-root node v (including the leaf nodes), real numbers γ_{vw} for each non-root ancestor w of v and real numbers $U_v(s)$ for each element $s \in S_v \times \prod_{w \in B_v} S_w$. From these we define a utility function u_v, which is a sum of two terms: $U_v(\sigma_{v,B_v})$, a multilinear function of the strategies executed by v and its siblings in B_v, and $\sum_{\text{nonroot ancestors } u} \gamma_{vw} u_w$. We require that the utility function u_v at a leaf node v be equal to the utility function u_i of the player i corresponding to the leaf node v.

We will refer to an emergent node tree structure as an *ENT* for short. Note that for a given normal form game, we can always define a class of ENTs by defining a tree with a single emergent node (the root node), so that all the leaf nodes are siblings. We call such an ENT *trivial*. For any given normal form game, there need be no nontrivial ENT, or there may be many distinct possible ENTs.

The behavior of the *emergent players* is completely determined by the behavior of the actual players (the leaf nodes). The *emergent strategy* σ_v executed by the emergent player v when the actual players execute strategy profile σ is defined recursively by multilinearity:

$$\sigma_v(s_{vk}) = \sum_{s \in S_{C_v}} p_v(k, s) \prod_{w \in C_v} \sigma_w(s_w).$$

So we compute the emergent strategies from the bottom up.

From the above definition, we see that at a non-root node w of the tree, the utility function is

$$\begin{aligned} u_w(\sigma) &= U_w(\sigma_w) + \sum_{\text{nonroot ancestor } v} \gamma_{wv} u_v(\sigma) \\ &= \sum_{s \in S_w \times \prod_{x \in B_w} S_x} U_w(s)\, \sigma_w(s_w) \prod_{x \in B_w} \sigma_x(s_x) + \sum_{\text{nonroot ancestor } v} \gamma_{wv} u_v(\sigma). \end{aligned}$$

So we compute the utility from the top down.

We see that the utilities of each actual player (the leaf nodes) may depend on the strategies executed by every other actual player. So, the graphical model of the actual game may be the complete graph. Imposing an emergent node tree structure, corresponds to deleting some of these edges and adding more nodes, and edges connected to those nodes, to the graph, so that the new graph has a nontrivial structure. With the addition of the new variables $\sigma_v(s_{vk})$, we get more information about the sparsity of our multilinear equations.

In our definition, we did not require that the numbers γ_{vw} have the same sign for all descendants v of a node w. Thus, our definition does not require that the emergence of a node represent a common interest among its descendant nodes (although of course it does cover that situation).

For example, consider a normal form game with the ENT in Figure 3.5 where the leaf nodes correspond to

1. An American citizen

2. A Soviet saboteur living in America

3. A Soviet citizen

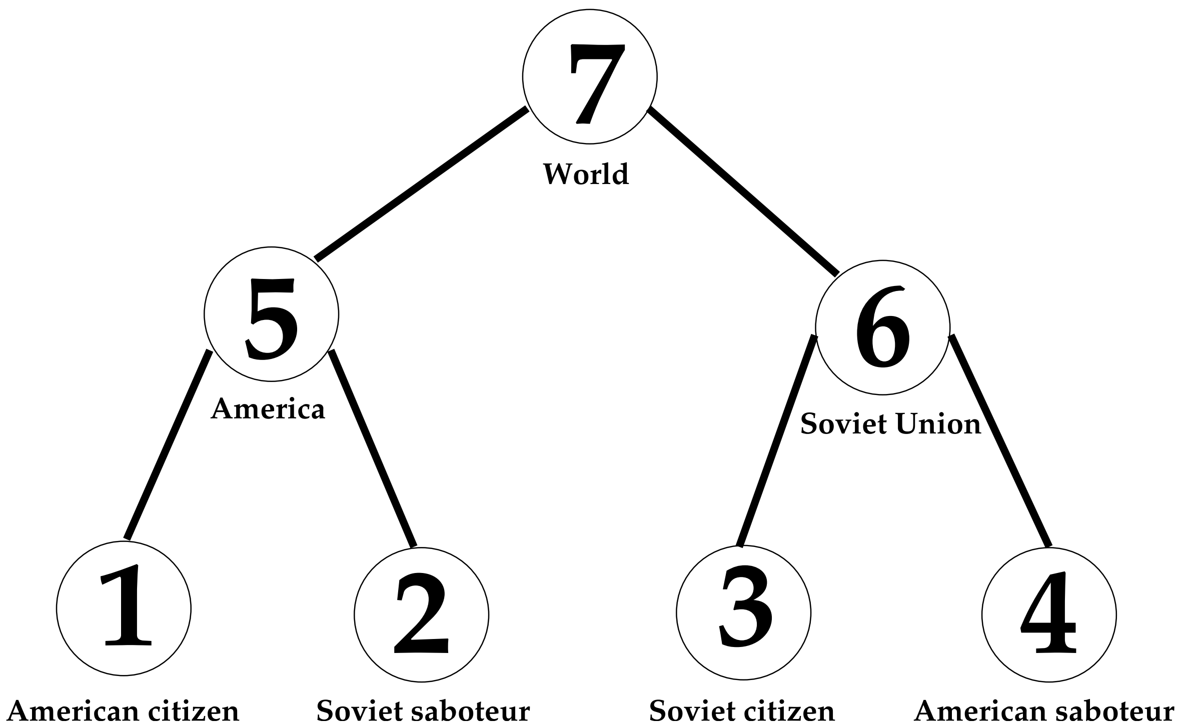

Figure 3.5: Emergent Node Structure For The Saboteur Game

4. An American saboteur living in the USSR

The parent of nodes 1 and 2 is node 5, corresponding to America, the parent of nodes 3 and 4 is node 6, corresponding to the USSR, and the the root is node 7, corresponding to the world. Then while $\gamma_{15} > 0$ and $\gamma_{36} > 0$, we have $\gamma_{25} < 0$ and $\gamma_{46} < 0$.

We now define a natural refinement of the equilibrium concept for games with an ENTs.

Definition. If a normal form game has an ENT as defined above, then a Nash equilibrium σ of that game is *hierarchically perfect* with respect to this ENT if for every emergent node v, given the strategies induced on the siblings of v by σ, the payoff $u(v)$ at v cannot be increased by changing only $\sigma(v)$.

Note that since our definition requires the linear map from the strategy space of the children of v to the strategy space of v to be full-rank, any strategy $\sigma'(v)$ deviating from $\sigma(v)$ which could result in a higher payoff $u(v)$ would be achievable by some strategy profile of the descendants of v.

We will also need the following definition:

Definition. A strategy profile of a normal form game with an ENT is *totally mixed with respect to this ENT* if it is totally mixed in the usual sense and the emergent strategies at each emergent node are also totally mixed.

Theorem 16. *For a generic game with an ENT as above, construct a directed graphical model G whose nodes are the nodes of the tree except the root, with edges as follows: the children in T of a node v form a directed clique in G, and each such child also has a directed edge from v and each ancestor of v except the root, and from each of their siblings. Then the Bernstein number we obtain by applying Theorem 13 to this directed graphical model is an upper bound on the number of totally mixed Nash equilibria of this game which are hierarchically perfect and totally mixed with respect to this ENT.*

Proof. This is the graphical model we would obtain if all the emergent players were actual players. That is, we have ignored the equations

$$\sigma_v(s_{vk}) = \sum_{s \in S_{C_v}} p_v(k, s) \prod_{w \in C_v} \sigma_w(s_w).$$

So the set of totally mixed Nash equilibria of our game which are hierarchically perfect with respect to this ENT is a subset of the set of totally mixed Nash equilibria of the game with this graphical model. □

Generically, there may be no hierarchically perfect totally mixed Nash equilibria. If the system of equations defining the quasiequilibria of the game with the directed graphical model is 0-dimensional, then none of the finitely many solutions to this system may satisfy the additional equations

$$\sigma_v(s_{vk}) = \sum_{s \in S_{C_v}} p_v(k, s) \prod_{w \in C_v} \sigma_w(s_w).$$

For example, consider a game as in Figure 3.5 in which each actual player has two pure strategies and each emergent player also has two pure strategies. Generically, a game with 4 players, each with 2 pure strategies, would have

$$\operatorname{per}\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = 9$$

quasiequilibria. On the other hand, if the game has an ENT as in Figure 3.5, then the directed graphical model given by the theorem is as in Figure 3.4. Thus there is no more than

$$\operatorname{per}\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = 1$$

quasiequilibrium which is hierarchically perfect and totally mixed with respect to this ENT. Indeed this would hold whenever the ENT is a binary tree, that is, each non-leaf node has two children, and all siblings have the same number of pure strategies.

For example, say that if players 1 and 2 either both choose their 0th pure strategy or both choose their 1st pure strategy, then the emergent strategy of node 5 is s_{51},

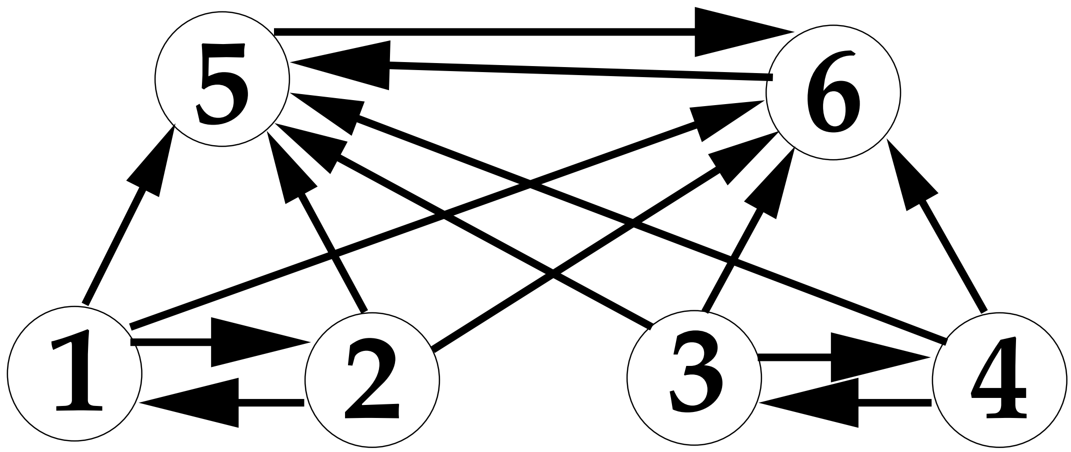

Figure 3.6: Graphical Model For The Saboteur Game

otherwise it is s_{50}. Similarly, if players 3 and 4 either both choose their 0th pure strategy or both choose their 1st pure strategy, then the emergent strategy of node 6 is s_{61}, otherwise it is s_{60}. Let U_5 and U_6 be given by

$$\begin{array}{cc} & \begin{array}{cc} s_{60} & s_{61} \end{array} \\ \begin{array}{c} s_{50} \\ s_{51} \end{array} & \begin{pmatrix} 0,0 & 0,-1 \\ 7,0 & -5,1 \end{pmatrix}, \end{array} \tag{3.2}$$

(where the (i,j)th entry is the pair $U_5(s_{5i}, s_{6j}), U_6(s_{5i}, s_{6j})$). Let $\gamma_1 = \gamma_3 = 1$ and $\gamma_2 = \gamma_4 = -1$. Let U_1 and U_2 be given by

$$\begin{array}{cc} & \begin{array}{cc} s_{20} & s_{21} \end{array} \\ \begin{array}{c} s_{10} \\ s_{11} \end{array} & \begin{pmatrix} 0,0 & 0,-1 \\ 1,0 & -4,1 \end{pmatrix}, \end{array} \tag{3.3}$$

and let U_3 and U_4 be given by

$$\begin{array}{cc} & \begin{array}{cc} s_{40} & s_{41} \end{array} \\ \begin{array}{c} s_{30} \\ s_{31} \end{array} & \begin{pmatrix} 0,0 & 0,-1 \\ 1,0 & -3,2 \end{pmatrix}. \end{array} \tag{3.4}$$

We abbreviate $\sigma_i(s_{i1})$ as σ_i by abuse of notation. At a totally mixed Nash equilibrium σ which is hierarchically perfect and totally mixed with respect to the ENT of Figure

3.5, we have $0 = U_5(s_{50}, \sigma_6) = U_5(s_{51}, \sigma_6) = 7(1 - \sigma_6) - 5\sigma_6 = 7 - 12\sigma_6$, so $\sigma_6 = \frac{7}{12}$. Similarly we have $0 = -(1 - \sigma_5) + \sigma_5 = 2\sigma_5 - 1$ so $\sigma_5 = \frac{1}{2}$.

We also have $u_1(s_{10}, \sigma_2, \sigma_5, \sigma_6) = U_1(s_{10}, \sigma_2) + u_5(\sigma_5, \sigma_6)$, which we must equate to $u_1(s_{11}, \sigma_2, \sigma_5, \sigma_6) = U_1(s_{11}, \sigma_2) + u_5(\sigma_5, \sigma_6)$, for hierarchical perfection (here we are ignoring the fact that σ_5 is a function of σ_1 and σ_2). This gives us that $0 = U_1(s_{10}, \sigma_2) = U_1(s_{11}, \sigma_2) = (1 - \sigma_2) - 4\sigma_2 = 1 - 5\sigma_2$, so $\sigma_2 = \frac{1}{5}$. Similarly we have $u_2(s_{20}, \sigma_1, \sigma_5, \sigma_6) = U_2(s_{20}, \sigma_1) - u_5(\sigma_5, \sigma_6)$, which we must equate to $u_2(s_{21}, \sigma_1, \sigma_5, \sigma_6) = U_2(s_{21}, \sigma_1) - u_5(\sigma_5, \sigma_6)$, so $U_2(s_{20}, \sigma_1) = U_2(s_{21}, \sigma_1)$. This gives $0 = -(1 - \sigma_1) + \sigma_1 = 2\sigma_1 - 1$, so $\sigma_1 = \frac{1}{2}$. We also have $0 = (1 - \sigma_4) - 3\sigma_4 = 1 - 4\sigma_4$, so $\sigma_4 = \frac{1}{4}$, and $0 = -(1 - \sigma_3) + 2\sigma_3 = 3\sigma_3 - 1$, so $\sigma_3 = \frac{1}{3}$.

Finally, we check that $\sigma_1\sigma_2 + (1 - \sigma_1)(1 - \sigma_2) = \frac{1}{10} + \frac{4}{10} = \frac{1}{2} = \sigma_5$, and $\sigma_3\sigma_4 + (1 - \sigma_3)(1 - \sigma_4) = \frac{1}{12} + \frac{6}{12} = \frac{7}{12} = \sigma_6$. Now given σ_{-1}, player 1 cannot increase either U_1 or u_5 by changing only σ_1, so player 1 cannot increase u_1. Similarly, player 2 can neither increase U_1 nor *decrease* u_5 by changing only σ_2, so player 2 cannot increase u_2. In this way, we see that σ is a Nash equilibrium of the actual game.

A strategy profile of the actual players is a point in the product of probability simplices corresponding to their actual strategy spaces. When we pass to an emergent player one level up, we project the product of simplices for the actual players below that emergent player to a smaller dimensional simplex, the space of emergent mixed strategies of this emergent player. That we are able to do this means that the payoffs to other actual players, not below this emergent player, depend only on the choice of a point in the smaller dimensional simplex by these actual players.

We can use ENTs to analyze certain cooperative games. We consider each coalition to be an emergent player. An actual player's pure strategies specify the highest level of coalition to join. So the number of its pure strategies is the number of its ancestors in the tree(including itself). Each coalition forms if all its descendants agree to join it, otherwise it doesn't form. The number of pure strategies of a coalition is one more than the number of its ancestors (including itself). Its pure strategies correspond either to the highest level of coalition containing this coalition which its members have agreed to form, or to not forming this coalition itself. The function U_v for each coalition v is zero if the coalition forms and is equal to the *value* of the coalition if it does form; it does not depend on the actions of v's siblings. The number γ_{vw} represents v's share of the gain from the larger coalition w, if it forms.

Note that a given ENT does not allow all possible subsets of players to form coalitions, but only certain ones. We could extend the definition to all possible subsets by positing that for any partition of a coalition into subcoalitions not in the tree, the subcoalitions receives the same utility by joining or not joining the coalition. Thus not all cooperative games correspond to ENTs. Those that do, however, may often occur in modeling real situations.

Chapter 4

Tools For Computing Nash Equilibria

4.1 Introduction

The main tool for computing Nash equilibria today is the free software package Gambit of McKelvey, McLennan, and Turocy [MMT]. It implements a variety of techniques for finding a single Nash equilibrium of a game, and a single technique for finding all the Nash equilibria. Since finding all the Nash equilibria is difficult, they are not often computed in practice. However, techniques for solving systems of polynomial equations have continued evolving for several years since their implementation in Gambit. Here we experiment with the use of various general-purpose polynomial system solvers to solve polynomial equations arising from games, and thus find all the Nash equilibria. Our goal is to determine which of the algebraic techniques today performs best for these problems. We find that the polyhedral homotopy continuation package PHC of Verschelde is robust and able to solve games with thousands of Nash equilibria. Furthermore, polyhedral homotopy continuation clearly lends itself to parallelization, although we have not had the opportunity to experiment with this.

We concentrate on the problem of computing all totally mixed Nash equilibria. Recall that once we have a procedure to do this, it can be used as a subroutine to compute all Nash equilibria. For every subset of the set of all pure strategies of all players not containing all of any particular player's pure strategies, one derives a new normal form game in which that subset of pure strategies is unavailable, and finds the totally mixed Nash equilibria for the new game. Then one checks if these would still be Nash equilibria if the deleted pure strategies were available. If so, then these are partially mixed Nash equilibria of the original game, that is, they are equilibria in which the probabilities allocated to the pure strategies in the subset are zero. In the special case when for each player, the subset contains all but one of that player's strategies, the resulting point is trivially a Nash equilibrium in the new game, and potentially a pure strategy Nash equilibrium of the old game. This is in fact how Gambit computes all Nash equilibria of a game with more than two players. (There are many other techniques to find a *single* Nash equilibrium. However, none of these techniques, even when repeated, can be guaranteed to find all Nash equilibria. Moreover, the single Nash equilibrium found in this way is not distinguished by any special properties which would make it more significant in the analysis of the game.)

We have seen that the problem of computing totally mixed equilibria reduces to that of solving a system of polynomial equations subject to some inequality constraints. As we saw in Chapter 2, the solution set can be (stably isomorphic to) any algebraic variety (i.e., to the solution set of any polynomial system). However, Harsanyi [Har73] showed that the set of payoffs for which there are finitely many Nash equilibria is a generic set. In his formulation "generic" meant "except on a set of measure zero". From an algebraic point of view, this implies that for any assignment of payoff values outside of an algebraic subset of positive codimension in the space of all such assignments, the solution set to the polynomial system is a zero-dimensional algebraic variety. This chapter focuses on applying various techniques to compute the complete set of totally mixed Nash equilibria.

4.2 The Status Quo: Gambit

Gambit, developed by McKelvey, McLennan, and Turocy, is currently the standard software package for computing Nash equilibria. Most of the code focuses on solving two-person games. (Indeed, a large proportion of the game theory literature itself focuses on two-person games. This may be because the two-person situation is already quite rich and interesting, but it also may be partially due to the current inability to solve even moderate examples of games with more than three players.) For normal-form games with more than two players, the program can compute all equilibria with a routine called `PolEnumSolve`. It can also compute single equilibria (or multiple equilibria, one at a time, with no guarantee at any point that all have been found) with several other algorithms. However, since the algebraic techniques (including `PolEnumSolve`) which we are comparing here solve for all equilibria, we will only consider `PolEnumSolve`. By default, `PolEnumSolve` solves for all Nash equilibria by recursing over the possible subsets of used strategies as explained above. But we chose not to recurse for ease of comparison with the other methods, which we will use to compute all (and only) the totally mixed Nash equilibria.

`PolEnumSolve` works by spatial subdivision, a technique often used as well in computer-aided geometric design. The algorithm starts with a higher-dimensional cube that contains the entire strategy space. It uses Newton iteration to find a solution within this cube. If there is one, it checks whether this solution is within the strategy space; if not it discards it and starts again. When it finds a bona fide solution, it checks that it is the unique solution within this cube—in fact, within the sphere circumscribing the cube. Basically, if the system were linear it could have no other solutions at all (since the solution set is zero-dimensional). Gambit checks that the linear Taylor approximation is good enough (that is, the nonlinear part of the system is small enough) within this sphere to guarantee that there is no other root inside. If the solution cannot be guaranteed to be unique, then the cube is subdivided and the process is repeated within the smaller cubes.

We generated various games with random entries to try to solve with Gambit. (We used the standard Ocaml library routines to generate random numbers uniformly distributed within a fixed range.) Unfortunately the current version of Gambit (version 0.97.0.3, "Legacy"), is extremely unstable and crashes with a segmentation fault on many of the simplest games. The only games which it was able to solve with any consistency were the smallest case of more than two players, namely three players each with two pure strategies. (Even here many segmentation faults occurred.) These took it from 60 to 160 ms to solve. (All these computations, and all others reported here, were done on the same machine: a Dell Latitude C840 2.0GHz Mobile Pentium

4 laptop with 1GB RAM running Linux kernel version 2.4.19.)

4.3 Pure Algebra: Gröbner Bases

The set of solutions to a system of polynomial equations is called an *algebraic variety*. Conversely, consider the set of polynomials which vanish on a set of a points. Any sum of two such polynomials will also vanish on the same set, and so will any product of such a polynomial with any other polynomial. These two conditions mean that the set of polynomials vanishing on a set is an *ideal*, i.e., closed under addition and under multiplication by a polynomial.

A *generating set* for an ideal is a set of elements such that every other element is a sum of products of these elements with other polynomials. It so happens that every polynomial ideal has a finite generating set. Given such a generating set, one might try to determine whether a particular candidate polynomial lies in the ideal by dividing by the polynomial generators. However, in general, the remainder is not uniquely determined, and may not be zero even though the candidate polynomial lies in the ideal. This state of affairs is remedied by requiring that the generating set satisfy certain technical conditions (Buchberger's criterion). If it does, it qualifies as a *Gröbner basis*.

A Gröbner basis is defined with respect to a particular *term order*. There is a natural partial order on monomials, namely that induced by divisibility (with 1 being the least monomial). A *term order* extends this partial order to a total order, while respecting multiplication. More precisely, if $m_1 \prec m_2$ for a term order $\prec$, then $mm_1 \prec mm_2$ for any monomial m. Every polynomial ideal has a Gröbner basis with respect to each term order. If we specify that the Gröbner basis must be *reduced*, then it is unique for a given term order. (This means that no term in any element of the basis can be divisible by the leading term of another element of the basis.) Gröbner bases can be used to solve many of the fundamental problems of computational algebra.

Perhaps the most intuitive term order is the *lexicographic* one. One specifies an ordering of the variables. Then in comparing two monomials, one first compares the powers of the heaviest (greatest) variable. If they are unequal, this is decisive;

otherwise one compares the powers of the next heaviest variable, and so on.

The reduced Gröbner basis with respect to the lexicographic term order almost always has higher degrees than the Gröbner basis with respect to some other term orders, and so the computational complexity of many algorithms is worsened when using this order. However, this term order in particular supports solving 0-dimensional polynomial systems. It follows from elimination theory that from the reduced Gröbner basis, a collecton of triangular sets describing the solutions can be computed. A triangular set consists of a polynomial in which only one variable occurs, one in which that and another variable occur, one in which those two and a third variable occur, and so on. The roots of the first polynomial can be found numerically (or by a combination of symbolic and numerical methods). Then each of these values can be substituted into the second polynomial, making it a polynomial only in the second variable. Solving this numerically gives the values of the second variable, and so on.

Two popular software packages for Gröbner basis computations are Macaulay2 [GS] and Singular [GPS01]. Unfortunately Macaulay2 is not currently well set up for solving polynomial systems, although support is planned for the future. In our tests it took about 10 ms to find a Gröbner basis for the case of 3 players with 2 strategies each, and 2.64 seconds to find a Gröbner basis for the case of 4 players with 2 strategies each. On larger instances it exited with a segmentation fault. Of course, this does not include the time to actually use the Gröbner basis to find the roots, which would require exporting the problem to another numerical solving routine such as one from the Netlib repository or in Matlab. This is not trivial, since Macaulay2 computes Gröbner bases with arbitrary-precision coefficients (and indeed, the coefficients can rapidly become very large), whereas numerical solvers usually compute in fixed-precision.

Singular, on the other hand, did much better. It comes with a standard library `solve.lib` for complex symbolic-numeric polynomial solving. Using the main routine `solve` from this library, we were able to solve the case of 3 players with 2 strategies each in 10ms, and with 3 strategies each in 1150 ms. The case of 4 players with 2 strategies each was solved in 70 ms.

While the above results seem promising, they are eclipsed by the performance of the polyhedral homotopy continuation method, to which we turn next.

4.4 Polyhedral Homotopy Continuation: PHC

The polynomial systems we want to solve are very *sparse*. That is, given the total degree of each equation, we don't see all the terms that there could be in an equation of that degree. Specifically, these monomials are multilinear. Moreover, in the equation associated with player i, none of the variables associated with i appear, and in each term of such an equation, only one of the variables associated with any other particular player j can appear at a time.

The number of solutions to such a sparse 0-dimensional polynomial system is generically far smaller than the Bézout number obtained by multiplying total degrees. The sparsity of the system can be described in terms of the exponent vectors occurring in the monomials which occur in each polynomial. For a single polynomial, the convex hull of these vectors forms a lattice polytope, called its *Newton polytope*. The Minkowski sum of two polytopes is obtained by translating one of them by each of the vectors in the other and taking the convex hull of the result. Similarly, the Minkowski sum of any finite number of polytopes can be defined inductively. One can subdivide the Minkowski sum (non-uniquely) into smaller lattice polytopes by following the course of the various faces during the translation. This results in a *mixed subdivision*: each of its elements, called *cells*, is full-dimensional and is a Minkowski sum of faces from all the original polytopes. If a cell is the Minkowski sum of an edge or a vertex from each of the original polytopes, then it is a *mixed cell*. The mixed volume of the system is the sum of the normalized (with respect to the integer lattice) volumes of these mixed cells, and it is equal to the number of roots of a generic polynomial system with those Newton polytopes. This number is known as the *Bernstein number* after Bernstein, who proved this equality [Ber75].

This number is only a function of the Newton polytopes. The *polyhedral homotopy continuation* method, introduced by Huber and Sturmfels [HS95], takes advantage of this. In general, one uses homotopy continuation to solve a polynomial system by first starting with another system of the same multidegree whose roots are obvious (and all distinct—roots with multiplicity are generally troublesome for numerical solvers), and gradually perturbing (or "morphing") the coefficients towards the system of interest.

At each step, one finds all the roots of the intermediate system by iterating from the roots of the system in the previous step. In this way one traces out a path from each root of the starting system to each root of the system of interest (which is why this method is also called “path-following”). The main practical drawback previously was that many of these paths would not lead to roots, because the starting system was generic and *dense*, whereas polynomial systems arising in practice are usually sparse. Thus the number of paths would explode with the size of the problem. However, with polyhedral homotopy continuation, the starting system is also chosen to be sparse in the same way as the system of interest. So only those paths which can lead to actual roots are followed. Indeed, Bernstein originally proved his theorem by exhibiting a similar homotopy [Ber75], and a solver using this homotopy was previously implemented by Verschelde, Verlinden, and Cools [VVC94].

Verschelde’s software package PHC [Ver99] for polyhedral homotopy continuation is in continuous development yet is very stable. Furthermore, it is well-documented and very simple to use. We were able to solve the following cases:

- 3 players with 2 pure strategies each: 2 roots found in 20ms.
- 3 players with 3 pure strategies each: 10 roots found in 350ms.
- 3 players with 4 pure strategies each: 56 roots found in 13s280ms.
- 3 players with 5 pure strategies each: 346 roots found in 3m19s540ms.
- 3 players with 6 pure strategies each: 2252 roots found in 48m41s870ms.
- 4 players with 2 pure strategies each: 9 roots found in 260ms.
- 4 players with 3 pure strategies each: 297 roots found in 4m3s220ms.
- 4 players with 4 pure strategies each: 13833 roots found in 7h2m20s780ms.
- 5 players with 2 pure strategies each: 44 roots found in 7s200ms.
- 6 players with 2 pure strategies each: 265 roots found in 7m10s790ms.

The running time seems to go up somewhat superlinearly with the Bernstein number (which may be considered part of the inherent complexity of the problem). Furthermore, this method is trivially parallelizable, requiring no communication between processors following different paths. (In fact, quite recently an alternative implementation of polyhedral homotopy continuation, PHoM, including a parallel implementation, has been made available by Gunji, Kim, Kojima, et. al. [GKK+03].) For all but the smallest problems, it is the path-following which takes up most of the running time. Indeed, these polynomial systems are multilinear, which is a special case of *multihomogeneity*, and the mixed volume computations needed to set up the start system can be carried out more efficiently for multihomogeneous polynomial systems than in general. For the smallest games, though, the time used to compute the start system is significant. The start system, which depends only on the Newton polytope and thus can be the same for all games of a given format, could be precomputed.

4.5 Other Directions

An interesting direction for further work is the computation of Nash equilibria under uncertainty. Specifically, the payoff functions may not be known exactly, but only approximately. A natural formulation is that each payoff value is known to lie in some interval. This leads to the question of how the set of Nash equilibria varies as the set of payoff values (now considered as parameters) varies.

Purely symbolically, such variations could be studied either through parametric Gröbner bases, as computed for example by Montes [Mon02] or Faugère [Fau02], or through resultants, as computed for example by Emiris and Canny [EC95]. For parametric Gröbner bases, the basic idea is to carry out the Gröbner basis computation, treating the payoff values as parameters and assuming that no cancellations ever occur. In this way one arrives at the generic solution. (A cancellation occurs whenever two algebraic expressions involving the parameters are equal. Thus, if desired, one can keep track of the algebraic equations assumed not to hold along the way, and thus determine in the end the algebraic subvariety of the payoff space which is *not* generic, as a by-product of this computation.) One might define resultants as the end result of such computations, but in fact resultants can be expressed much more compactly using determinantal formulas. In either case, if such formulas were precomputed for games of various formats, the Nash equilibria for any specific set of payoffs could be computed by evaluation of the formula in polynomial time (provided the payoffs were indeed generic; if not, division by zero would occur). Such formulas have been computed by Emiris in his thesis [Emi94] for very small cases, but unfortunately for larger cases, these computations are still intractable for the present.

A recent approach to finding real roots of polynomial systems is through semidefinite programming. Semialgebraic constraints can include nonnegativity constraints (such as arise in our problem) as well as equations. These nonnegativity constraints are relaxed to the (sufficient) condition that the polynomials in question be the sums of squares of other polynomials (which of course are of lower degree). This condition can be expressed as the positive semidefiniteness of a matrix, namely the Gram matrix, which represents the quadratic form in the smaller polynomials in the monomial basis. Unfortunately we were not able to test the primary exemplars of this approach, SOStools and Gloptipoly, which require particular versions of Matlab. However, it is clear that this approach lends itself easily to the formulation of such "robust" computations. One uses parametric values for the payoffs and adds the constraints on the payoffs (for example, that they lie in a certain interval) to the problem.

As was discussed earlier, PHC succeeds at finding all the "quasi-equilibrium" points, and the result of McKelvey and McLennan [MM97] shows that these may all be actual Nash equilibria. Thus, there is no way to avoid worst-case complexity given by the Bernstein number. However, in practice it will often be the case that many of the "quasi-equilibrium" points do not lie in the product of simplices, or are not even real. Practically, time would be saved by heuristic methods for examining the starting system and determining that a significant subset of the paths will not converge to a solution in the product of simplices and so do not need to be followed. Such heuristic methods have yet to be defined.

Game theorists generally would prefer that there be one distinguished equilibrium point for any particular game. Not only does this allow the game theorists to predict what will happen during the game, it allows the players themselves to predict what the other players will do. After all, if different players have different equilibrium points in mind when choosing their strategies, then the resulting behavior may not even be at equilibrium. For this reason various refinements of the Nash equilibrium concept have been proposed; these are summarized in [vD87]. In the past algebraic techniques have been used to find all Nash equilibria, and other techniques have been used to try to find a single Nash equilibrium (preferably the "best" one in one of these senses). It would be interesting to see if the methods we have used can be modified to compute these more refined equilibria.

4.6 Conclusion

Game theory is a mathematical model of strategic interaction. The main computer package for studying game theory today is Gambit. Although there are many ways to characterize Nash equilibria, the one which lends itself most easily to the computation of *all* Nash equilibria of a game with more than two players is as solutions to systems of polynomial equations. However, the algorithm currently implemented in Gambit could be outperformed by the existing polyhedral homotopy continuation software PHC. So hopefully PHC or some similar package will soon be incorporated into Gambit. Furthermore, there are many other promising directions to pursue in applying algebra, and in particular computer algebra, to game theory.

Bibliography

[AK92] Selman Akbulut and Henry C. King. *Topology of real algebraic sets*. Springer-Verlag, 1992.

[Ber75] David N. Bernstein. The number of roots of a system of equations. *Functional Analysis and Applications*, 9(2):183–185, 1975.

[BG54] David Blackwell and M. Abraham Girshick. *Theory of Games and Statistical Decisions*. John Wiley & Sons, New York, 1954.

[Bor21] Émile Borel. La théorie du jeu et les équations, intégrales à noyau symmetrique gauche. *Comptes Rendus de l'Académie des Sciences*, 173(December):1304–8, 1921.

[Bor24] Émile Borel. Sur les jeux où interviennent l'hasard et l'habilité des joueurs. In *Théorie des probabilités*. J. Hermann, Paris, 1924.

[Bor27] Émile Borel. Sur les systèmes de formes linéaires à déterminant symmétrique gauche et la théorie générale du jeu. *Comptes Rendus de l'Académie des Sciences*, 184:52–53, 1927.

[CLO97] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, 1997.

[Dat03a] Ruchira S. Datta. Universality of Nash equilibria. *Mathematics of Operations Research*, 28(3):424–32, 2003.

[Dat03b] Ruchira S. Datta. Using computer algebra to compute Nash equilibria. *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 74–79, August 2003.

[EC95] Ioannis Emiris and John Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *Journal of Symbolic Computation*, 20(2):117–149, 1995.

[Emi94] Ioannis Emiris. *Sparse Elimination and Applications in Kinematics*. PhD thesis, University of California, Berkeley, December 1994.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Groebner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (ISSAC 2002)*, pages 75–83, 2002.

[GKK+03] Takayuki Gunji, Sunyoung Kim, Masakasu Kojima, Akiko Takeda, Katsuki Fujisawa, and Tomohiko Mizutani. PHoM—a polyhedral homotopy continuation method for polynomial systems. Research Report B-386, Tokyo Institute of Technology, Tokyo, January 2003. `http://www.is.titech.ac.jp/~kojima/PHoM/`.

[GPS01] Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. SINGULAR 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001. `http://www.singular.uni-kl.de`.

[GS] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at `http://www.math.uiuc.edu/Macaulay2/`.

[Har73] John Harsanyi. Oddness of the number of equilibrium points: a new proof. *International Journal of Game Theory*, 2:235–250, 1973.

[HS95] Birkett Huber and Bernd Sturmfels. A polyhedral method for solving sparse polynomial systems. *Mathematics of Computation*, 64:1541–1555, 1995.

[KL93] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61:1019–1045, 1993.

[KLS01] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory. *Uncertainty in artifical intelligence: proceedings of the seventeenth conference*, pages 253–260, 2001.

[Kou76] Anatoli G. Kouchnirenko. Newton polytopes and the Bezout theorem. *Functional Analysis and Applications*, 10:233–235, 1976.

[Kre90] David M. Kreps. *A Course in Microeconomic Theory*. Princeton University Press, Princeton, 1990.

[McL99] Andrew McLennan. The maximum number of real roots of a multihomogeneous system of polynomial equations. *Beiträge zur Algebra und Geometrie*, 40(2):343–350, 1999.

[MM96] Richard McKelvey and Andrew McLennan. Computation of equilibria in finite games. In H. Amman, D. A. Kendrick, and J. Rust, editors, *The Handbook of Computational Economics, Vol. I*, pages 87–142. Elsevier, 1996.

[MM97] Richard McKelvey and Andrew McLennan. The maximal number of regular totally mixed Nash equilibria. *Journal of Economic Theory*, 72:411–425, 1997.

[MMT] Richard D. McKelvey, Andrew McLennan, and Theodore Turocy. Gambit: Software tools for game theory. Available at `http://www.hss.caltech.edu/gambit/`.

[Mnë88] Nikolai E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 527–544. Springer, Heidelberg, 1988.

[Mon02] Antonio Montes. A new algorithm for discussing Groebner bases with parameters. *Journal of Symbolic Computation*, 33(2):183–208, 2002.

[Nas50] John Nash. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36:48–49, 1950.

[Nas52] John Nash. Real algebraic manifolds. *Annals of Mathematics*, 116:133–176, 1952.

[OR94] Martin J. Osborne and Ariel Rubinstein. *A Course In Game Theory*. MIT Press, 1994.

[Pas63] Blaise Pascal. *Pascal: œuvres complètes*, chapter La Règle Des Partis. Seuil, 1963.

[RGZ95] Jürgen Richter-Gebert and Gunter M. Ziegler. Realization spaces of 4-polytopes are universal. *Bulletin of the American Mathematical Society*, 32:403–412, 1995.

[Sho91] Peter W. Shor. Stretchability of pseudolines is np-hard. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, pages 531–554. American Mathematical Society, 1991.

[Smi68] J. Maynard Smith. *Mathematical Ideas in Biology*. Cambridge University Press, 1968.

[Stu02] Bernd Sturmfels. *Solving Systems of Polynomial Equations*. American Mathematical Society, Providence, Rhode Island, 2002.

[SV00] Andrew J. Sommese and Jan Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *Journal of Complexity*, 16:572–602, 2000.

[Tog73] Alberto Tognoli. Su una congettura di Nash. *Annali della Scuola Normale Superiore di Pisa*, 27:167–185, 1973.

[vD87] Eric van Damme. *Stability and Perfection of Nash Equilibria*. Springer Verlag, Berlin, 1987.

[Ver99] Jan Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, 1999.

[VVC94] Jan Verschelde, Pierre Verlinden, and Ronald Cools. Homotopies exploiting Newton polytopes for solving sparse polynomial systems. *SIAM Journal of Numerical Analysis*, 31(3):915–930, 1994.

[Wal50] Abraham Wald. *Statistical decision functions*. John Wiley & Sons, New York, 1950.

[Wel95] Michael P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3):360–362, 1995.