

## APPENDIX B

---

### GAMETHEORY 'BIMATRIX'

*Computer? I don't even know 'er!*  
- The Great Web Canadianizer

The *Mathematica* package `GameTheory'Bimatrix'` implements all of the algorithms for equilibrium enumeration and selection discussed in the text. It is loaded into a *Mathematica* notebook with the command

```
<< GameTheory'Bimatrix';
```

This appendix describes the installation of the package, explains its exported functions and gives a brief account of the way in which Nash equilibria in symbolic form are determined.

#### B.1 INSTALLATION

Simply copy the directory `GameTheory` from the accompanying CD-Rom into the `AddOns/Applications` subdirectory of your *Mathematica* system. David Avis's vertex enumeration program `lrs` is also required. The executable `lrs.exe` has been included on the CD-Rom for Windows operating systems. It can be located anywhere that the operating system can find it, for example in the main `Windows` directory. On Linux/Unix systems, go to the web site

[cgm.cs.mcgill.ca/avis/C/lrs.html](http://cgm.cs.mcgill.ca/avis/C/lrs.html)

and follow the downloading and installation instructions. The binary executable that you will compile is `lrs` and it must similarly be located on your path.

#### B.2 FUNCTIONS

`BimatrixForm[A,B,Options]` displays the bimatrix for a two-person game with finitely many strategies using the convention that player 1 is the row and player

2 the column player. Player 1(2)'s payoffs are taken from the input matrix  $A(B)$  and placed lower left(upper right) in the cell for each strategy combination. The `Options` parameter takes the form `Labels->{rowlabels,columnlabels}`, where *rowlabels*, *columnlabels* are lists of strings which describe the players' pure strategies. The default is `Automatic`, in which case the labels are displayed in the form  $R_1, R_2 \dots$  and  $S_1, S_2 \dots$ . You can use explicit labels for just one player as well, for example `Labels->{Automatic, {aa,ab,bb}}`.

`Undominated[P,A]` returns `True` if the mixed strategy  $P$  in the bimatrix game  $\langle P, Q, A, B \rangle$  is not dominated. Mixed strategies  $Q$  for player 2 can be tested with `Undominated[Q,Transpose[B]]`.

`NashEquilibria[A]` calculates one Nash equilibrium (or saddle point) of the matrix game with payoff matrix  $A$  for player 1 by solving the equivalent linear program, see Chapter 3. The elements of  $A$  should be rational numbers. It returns the result in the form `{P*,H1*,Q*,H2*}`.

`NashEquilibria[A,B,Options]` calculates Nash equilibria for the bimatrix game  $\langle P, Q, A, B \rangle$ . It returns (with one exception, see below) the list `{P*,H1*,Q*,H2*}, \dots`. The elements of the matrices  $A$  and  $B$  can be rational numbers, symbolic expressions or a mixture of the two. The options are `Select` and `Symbolic`.

`Select->Automatic` (default) determines all of the equilibria of a non-degenerate game using the support enumeration method described in Chapter 2.

`Select->All` determines all extreme equilibria of a bimatrix game using the vertex enumeration method of Chapter 4.

`Select->One` determines one equilibrium with the Lemke-Howson algorithm described in Appendix A.

`Select->QS` determines all quasi-strict extreme equilibria.

`Select->ESS` determines all evolutionarily stable extreme equilibria of a symmetric bimatrix game which satisfy Haigh's criterion as described in Chapter 6.

`Select->Perfect` determines all perfect extreme equilibria, that is, all extreme equilibria which consist of undominated strategies (Chapter 7).

`Select->MNS` determines the maximal Nash subsets of a bimatrix game (Chapter 4). They are returned as the three-element list `{Ps,Qs,mns}`, where  $P_s$  and  $Q_s$  are lists of all of the players' extreme equilibrium strategies and  $mns$  is a list of connected components. Each such component is in turn a list of structures of form `{ps,qs}`, where the elements of  $ps$  index the list  $P_s$  and the elements of  $qs$  index the list  $Q_s$ . The convex hulls of the indexed strategies determine the maximal Nash subsets.

`Symbolic->Automatic` (default) All matrix elements of  $A$  and  $B$  are assumed to be rational numbers.

`Symbolic->substitutions`. Some elements of  $A$  and/or  $B$  contain symbolic expressions. The variable *substitutions* must then be a list of representative numerical values (rational numbers) for each symbol used. `NashEquilibria` returns the list `{P*,H1*,Q*,H2*}, \dots` in symbolic form.