

A Computer Program for the Computation of Equilibria of  
Finite Games and other Game  
Theoretical Computations

Group 20

Per Frost  
Marcus Lång  
Markus Thurlin  
Christer Hedberg  
Christopher Engelbrektsson



MethodName      getElement  
Class            LinearProgramSolution  
Parameters      An integer N  
Return value    A reference to a floating point number  
Description     Returns the N-th element of the vector that  
represents the  
                  solution to the linear programming problem from  
which  
                  the LinearProgramSolution is constructed  
Preconditions   N is nonnegative and less than the length of  
the vector      that  
                  represents  
Postconditions  None  
Called by       The constructor of LinearProgramSolution  
Calls           Nothing  
Validity checks Fails silently if the integer is out of  
range  
Accesses       Private solution vector

MethodName      LinearProgramSolution  
Class            LinearProgramSolution  
Parameters      A LinearProgram LP  
Return value    Inapplicable  
Description     Constructs a LinearProgramSolution from a  
linear program  
Preconditions   None  
Postconditions  None  
Called by       The constructors of NashEquilibrium and  
SubgamePerfectEquilibrium  
Calls           External functions, the member function  
                  getVector of LinearProgram and the member  
                  function getMatrix of LinearProgram  
Accesses       Private solution vector

MethodName      getVector  
Class            LinearProgram  
Parameters      None  
Return value    A vector of real numbers  
Description     Returns the vector that with a matrix  
represents a linear  
                  programming problem  
Preconditions   None  
Postconditions  None  
Called by       LinearProgramSolution  
Calls           Nothing  
Accesses       Private vector representing the linear program

MethodName      getMatrix  
Class            LinearProgram  
Parameters      None  
Return value    A matrix of real numbers

Description Returns the matrix that with a vector represents a linear programming problem  
Preconditions None  
Postconditions None  
Called by LinearProgramSolution  
Calls Nothing  
Accesses Private matrix representing the linear program

MethodName ParseError  
Class ParseError  
Parameters A String Str  
Return value A String containing information about the encountered exception  
Description Handles errors encountered during parsing

Preconditions A parsing attempt has been made  
Postconditions The exception has been sent to OutputStream.  
Called by The constructor of InputStream  
Calls OutputStream  
Accesses Nothing

MethodName GetParticularSolution  
Class Solution  
Parameters A Solution Solve  
Return value Stream containing solution  
Description Final handling of solutions  
Preconditions Specialised solution methods  
Postconditions Solution has been sent to OutputStream  
Called by Specialised solution methods  
Calls External functions, GetSolutionType and OutputStream  
Accesses Private solution vector

MethodName GetSolutionType  
Class Solution  
Parameters A String Str  
Return value An integer N  
Description Tells GetParticularSolution how to format output  
Preconditions Solution type has been given  
Postconditions GetParticularSolution knows how to format output  
Called by GetParticularSolution  
Calls None  
Accesses Nothing

MethodName read  
Class InputStream  
Parameters A reference to an istream object.

Return value A boolean.  
Description Returns true if and only if the InputStream object is successfully constructed  
Preconditions The istream object contains a representation of an InputStream.  
Postconditions The InputStream is valid.  
Called by The constructor of InputStream.  
Calls Methods of istream.  
Accesses Local variables of InputStream.

MethodName getOutputType  
Class InputStream  
Parameters None  
Return value An element of an enumeration of allowable output types.  
Description Returns the output type specified in the istream object from which the InputStream was constructed was.  
Preconditions The istream object contains a representation of an InputStream.  
Postconditions None  
Called by The constructor of ExtensiveFormGame and the constructor of NormalFormGame  
Calls Nothing  
Accesses Local variables of InputStream.

MethodName getInputType  
Class InputStream  
Parameters None  
Return value An element of an enumeration of allowable input types.  
Description Returns the input type specified in the istream object from which the InputStream was constructed was.  
Preconditions The istream object contains a representation of an InputStream.  
Postconditions None  
Called by The constructor of ExtensiveFormGame and the constructor of NormalFormGame  
Calls Nothing  
Accesses Local variables of InputStream.

MethodName getExtensiveFormGame  
Class InputStream  
Parameters None

Return value An object of type ExtensiveGame  
Description Returns an ExtensiveGame constructed from the  
InputStream from which the InputSteam was  
constructed.  
Preconditions The getInputType method returns a value  
consistent with that an ExtensiveGame can be  
constructed from the stream  
Postconditions None  
Called by None  
Calls The constructor of ExtensiveFormGame  
Accesses Nothing

MethodName getNormalFormGame  
Class InputStream  
Parameters None  
Return value An object of type NormalGame  
Description Returns an NormalGame constructed fromt the  
InputStream from which the InputSteam was  
constructed.  
Preconditions The getInputType method returns a value  
consistent with  
that an NormalFormGame can be constructed  
from the stream  
Postconditions None  
Called by None  
Calls The constructor of NormalFormGame  
Accesses Nothing

MethodName write  
Class OutputStream  
Parameters A pointer to a char and the number of bytes  
to be written.  
Return value A boolean.  
Description Returns true if writing to the output stream  
is successful,  
returns false otherwise.  
Preconditions There is a block of at least the length of  
the number of bytes  
to be written a the pointer.  
Postconditions The bytes are written to the ouput stream.  
Called by Functions overriding the function write in  
Solution  
Calls Nothing  
Accesses Local variables.

MethodName write  
Class Solution  
Parameters An OutputStream object.  
Return value A boolean.  
Description Returns true if the Solution is successfully  
written to the output stream, false otherwise. It is pure

virtual.  
Preconditions Inapplicable.  
Postconditions Inapplicable.  
Called by Inapplicable.  
Calls Inapplicable.  
Accesses Inapplicable.

MethodName write  
Class SubgamePerfectEquilibrium  
Parameters A reference to an OutputStream object.  
Return value A boolean.  
Description Returns true if the SubgamePerfectEquilibrium is successfully written to the output stream, false otherwise. It is pure virtual.  
Preconditions None  
Postconditions The OutputStream is properly constructed.  
Called by None  
Calls The write method of OutputStream.  
Accesses Local variables.

MethodName write  
Class NashEquilibrium  
Parameters A reference to an OutputStream object.  
Return value A boolean.  
Description Returns true if the NashEquilibrium is successfully written to the output stream, false otherwise. It is pure virtual.  
Preconditions None  
Postconditions The OutputStream is properly constructed.  
Called by None  
Calls The write method of OutputStream.  
Accesses Local variables.

## 5.6 Package diagram

