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

to verify a specified functionality of the system. Each test case must include the following:

functionality was originally described. functionality. effects of the tested functionality. system.

Define preferences for game states/payoffs for game states

Description

The system shall provide a method for the user to define preferences/payoffs for game states. This shall be tested by having some person associate a player with some game state.

Reference

Requirements Document, Section 6.1, Part **Define preferences for game states/payoffs for game states**

Inputs

A game state consists of atleast a payoff value.

Expected output

The expected output is a node that has atleast a value representing its payoff. For further definition of the game state see 'Associate players with game states'. The game state will be displayed as a node.

Step by step procedure

1. The user has the program open. 2. The user clicks the create game state button. 3. The user chooses a value for the gamestate. 4. The game state will be created and will be represented as a node on the screen.

Define information sets

Description

The system shall provide a method for the user to define information sets. This shall be tested by having any person define an information set.

Reference

Requirements Document, Section 6.1, Part **Define information sets**

Inputs

The information set is defined by a set of game states. The user does this by selecting nodes on the screen in the game tree editing tool.

Expected output

The expected output is a set of nodes as selected by the user. The result will be visible to the user in the graphical representation by connection with dashed lines between these nodes.

Step by step procedure

1. The user has defined gamestates on the screen. 2. The user chooses the tool for creating information sets. 3. The user selects the nodes he desires to be included in the set. 4. The user can observe the visible output of the changes to see if it is as desired.

Define player sets

Description

The system shall provide a method for the user to define player sets. This shall be tested by having some person define a player set.

Reference

Requirements Document, Section 6.1, Part **Define player sets**

Inputs

The player set consists of the players in a game, each player is defined simply by a name.

Expected output

A set of players who are included in a game.

Step by step procedure

1. The user clicks add player option. 2. The user enters a name for the player. 3. The player is added to the set or if there is no set a set is created for the game and the player is added. 4. The user will be able to see the player set as a list of players in a game including the added one.

Define transition between elements

Description

The system shall provide a method for the user to define transitions between elements. This shall be tested by having some person define a transition between some pair of two elements.

Reference

Requirements Document, Section 6.1, Part **Define transition between elements**

Inputs

The transitions between elements are simply the moves of the game, a tranistion from one gamestate to another. The input to this would be two nodes and the direction of the allowed move would be defined by the order in which the nodes are selected (as from 'first node' to 'second node').

Expected output

The expected output is a set of two gamestates. To the user this will be visibly observable by a connection line between the two gamestates with an arrow in the direction of the allowed move.

Step by step procedure

1. The user has 2 or more defined game states on the screen. 2. The user selects the tool for adding a transition. 3. The user selects two nodes that represent game states in the order he desired. 4. A transition will be defined between the two nodes and it will be displayed as an arrowed line between the nodes in the direction allowed.

Associate players with game states

Description

The system shall provide a method for the user to associate players with transitions. This shall be tested by having some person associate a player with some game state.

Reference

Requirements Document, Section 6.1, Part **Associate players with transitions**

Inputs

Associating a player with a game state requires a defined game state and a defined player.

Expected output

The expected output is a tuple consisting of a game state and a player. The association will be visible to the user.

Step by step procedure

1. The user has atleast one defined game state and atleast one defined player. 2. The user selects the game state and is able to see the value box. 3. The user can set the player parameter to any player that is defined in the game. 4. The user will see which player is associated with each transition by a lable at the game states node.

Tree-editing tool

Description

The system shall allow for taking graphical input and converting into a game tree. The tool will provide methods for creating the graphical input and use the required defined elements to create a game tree. The tree shall constantly be displayed regardless of its completeness. The user shall be able to store this tree to disk.

Reference

Requirements Document, Section 6.1, Part **Tree-editing tool**

Inputs

The game tree may consist of all elements described above.

Expected output

The expected output is a complete gametree in a clear visible format that is in a state it is ready to have the desired calculations applied.

Step by step procedure

1. The user defines all game states, tranitions, players and associates them as desired in a tree format. 2. The user clicks a button to 'complete the game'. 3. The system checks the suggested game tree and if anything is incomplete the user is informed and moved back to step one where he can complete the

tree. 4. The user is informed that the game is properly defined and the game is displayed with the options to perform calculations.

Calculate pure Nash equilibrium

Description

The system shall, given a valid input, calculate the pure Nash equilibria and return it in a requested output form. The user shall have the option of storing the result to disk.

Reference

Requirements Document, Section 6.1, Part **Calculate pure Nash equilibrium**

Inputs

A correctly defined game.

Expected output

The expected output will be a set of pure nash equilibria and will be displayed to the user as a set of sets of transitions.

Step by step procedure

1. The user has a correctly defined game. 2. The user clicks the option to the calculate pure Nash equilibrium. 3. The system will process the game and calculate its pure nash equilibria. 4. The output will be displayed to the user.

Calculate Nash equilibria in the mixed extension of a game (mixed Nash equilibria

Description

The system shall, given a valid input, calculate the mixed Nash equilibria and return it as requested output form. The user shall have the option of storing the result to disk.

Reference

Requirements Document, Section 6.1, Part **Calculate Nash equilibria in the mixed extension of a game (mixed Nash equilibria**

Inputs

A correctly defined game. For this calculation the user may define probability values for the transitions between game states.

Expected output

The expected output will be a set of mixed Nash equilibria and will be displayed to the user as a set of sets of transitions.

Step by step procedure

1. The user has a correctly defined game. 2. The user clicks the option to calculate the mixed Nash equilibrium. 3. The system will process the game and calculate its mixed Nash equilibria. 4. The output will be displayed to the user.

Calculate correlated equilibria

Description

The system shall, given a valid input, calculate the correlated equilibria and return it as requested output form.

Reference

Requirements Document, Section 6.1, Part **Calculate correlated equilibria**

Inputs

A correctly defined game.

Expected output

The expected output will be a set of correlated equilibia and will be displayed to the user as a set of sets of transitions.

Step by step procedure

1. The user has a correctly defined game. 2. The user clicks the option to calculate the correlated equilbrium. 3. The system will process the game and calculate its correlated equilibria. 4. The output will be displayed to the user.

Calculate sub game perfect equilibria

Description

The system shall, given a valid input, calculate the sub game perfect equilibria and return it as requested output form. The user shall have the option of storing the result to disk.

Reference

Requirements Document, Section 6.1, Part **Calculate sub game perfect equilibria**

Inputs

A correctly defined game.

Expected output

The expected outut will be a set of subgame perfect equilibria and will be displayed to the user as a set of sets of transitions.

Step by step procedure

1. The user has a correctly defined game. 2. The user clicks the option to calculate the sub game perfect equilibrium. 3. The system will process the game and calculate its sub game perfect equilibria. 4. The output will be displayed to the user.