This tutorial illustrates how to use the graphical interface provided by Gambit to construct and analyze a simple one-card poker game. This game is similar to one that appears in many texts, for example, Game Theory: Analysis of Conflict by Roger Myerson.
We'll explain the rules of the game as we go along. This is perhaps the simplest game that could be called "poker." It will be played between two players, whom we will call Alice and Bob. Let's begin by giving our game a descriptive title, and naming our players.
Chance

Player 1

Player 2

$(\omega)$
Game properties

Title: Untitled Extensive Form Game

Comment:

Information about this game:
- Filename:
- Number of players: 2
- This is a constant-sum game
- This is a game of perfect recall
Game properties

General
Title:Untitled Extensive Form Game

Players

Information about this game
Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

[Options: Cancel, OK]
Title: Untitled Extensive Form Game

Number of players: 2

This is a constant-sum game
This is a game of perfect recall
Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
Game properties

General

Title

Comment

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

Players

Cancel  OK
Chance

Player 1

Player 2

Game properties

<table>
<thead>
<tr>
<th>General</th>
<th>Players</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A sim</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
</tbody>
</table>

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
Game properties

General | Players
Title   | A simpl|

Comment

Information about this game
Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
Game properties

Title: A simpl

Comment:

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

Cancel  OK
Chance

Player 1

Player 2

Game properties

General | Players
---|---
Title | A simple
Comment

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

Cancel | OK
Chance

Player 1

Player 2

Game properties

General | Players
---|---
Title | A simple |
Comment

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

Cancel | OK
Game properties

Title: A simple ga

Comment:

Information about this game

Filename:

Number of players: 2

This is a constant-sum game

This is a game of perfect recall
Game properties

General | Players

Title: A simple game

Comment:

Information about this game

Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
Chance

Player 1

Player 2

Game properties

General | Players
Title: A simple game
Comment:

Information about this game
Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall

Cancel  OK
Title: A simple game

Information about this game:
- Filename:
- Number of players: 2
- This is a constant-sum game
- This is a game of perfect recall
Game properties

Title: A simple game

Information about this game:
- Filename:
- Number of players: 2
- This is a constant-sum game
- This is a game of perfect recall


**Game properties**

<table>
<thead>
<tr>
<th>General</th>
<th>Players</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A simple game of</td>
</tr>
</tbody>
</table>

**Comment**

- Information about this game
- Filename:
- Number of players: 2
- This is a constant-sum game
- This is a game of perfect recall
Chance

Player 1

Player 2

Game properties

General
Title: A simple game of

Players

Information about this game
Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
A simple game of perfect recall
A simple game of polo

This is a constant-sum game
This is a game of perfect recall
A simple game of poker

Information about this game:
- Filename: 
- Number of players: 2
- This is a constant-sum game
- This is a game of perfect recall
Game properties

General tab:
Title: A simple game of poker
Comment:

Players tab:
Information about this game:
Filename:
Number of players: 2
This is a constant-sum game
This is a game of perfect recall
The players' names are edited on the Players tab of this dialog.
Game properties

Players

t action about this game:
ne:
'f of players: 2
a constant-sum game
a game of perfect recall
### Game properties

<table>
<thead>
<tr>
<th>Player</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Player 1</td>
</tr>
<tr>
<td>2</td>
<td>Player 2</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>A</td>
<td>255,0,0</td>
</tr>
<tr>
<td>Player 2</td>
<td>0,0,255</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>1</td>
<td>Ali</td>
</tr>
<tr>
<td>2</td>
<td>Player 2</td>
</tr>
<tr>
<td></td>
<td>255,0,0</td>
</tr>
<tr>
<td></td>
<td>0,0,255</td>
</tr>
</tbody>
</table>
### Game properties

<table>
<thead>
<tr>
<th>Player</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice: 255,0,0</td>
</tr>
<tr>
<td>2</td>
<td>Player 2: 0,0,255</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
<td>Alice 255,0,0</td>
</tr>
<tr>
<td>2</td>
<td>Player 2 0,0,255</td>
</tr>
</tbody>
</table>
### Game Properties

<table>
<thead>
<tr>
<th>Player</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
</tr>
<tr>
<td>2</td>
<td>Player 2</td>
</tr>
</tbody>
</table>

**Color Values:**
- 1: Alice = (255, 0, 0)
- 2: Player 2 = (0, 0, 255)
### Game properties

<table>
<thead>
<tr>
<th>Player</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice 255,0,0</td>
</tr>
<tr>
<td>2</td>
<td>B 0,0,255</td>
</tr>
</tbody>
</table>
### Game properties

<table>
<thead>
<tr>
<th>Player</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
</tr>
<tr>
<td>2</td>
<td>Bo</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
<td>Alice</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
</tr>
<tr>
<td></td>
<td>255,0,0</td>
</tr>
<tr>
<td></td>
<td>0,0,255</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>1 Alice</td>
<td>255,0,0</td>
</tr>
<tr>
<td>2 Bob</td>
<td>0,0,255</td>
</tr>
<tr>
<td>Player</td>
<td>Color</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>1 Alice</td>
<td>255,0,0</td>
</tr>
<tr>
<td>2 Bob</td>
<td>0,0,255</td>
</tr>
</tbody>
</table>
The panel at left has updated to show the names of our protagonists. This panel lists the players in the game, and will summarize useful information about the equilibria we will compute.
Now let's build the game tree. The game will begin with Alice drawing a card from a deck. Let us suppose there are two types of cards: "high" cards and "low" cards, and that these are equally likely. (Perhaps, if this is a standard deck of cards, then red cards are high and black cards are low.)

To represent this, we will first create a chance, or nature, move. Click on the dice icon to the left of the word "Chance", and drag it to the root node of the tree.
\[ \omega \]
Chance

Alice

Bob

(ω)
We have created a chance move with two actions. The notation C:1 under the root node indicates that this is a chance move, and that it is move number 1 for the chance player.

The 1/2 beneath each branch indicates that the actions are equally likely. This is what we want. Gambit also gives the actions the default names "1" and "2".

Let's change those names to something more descriptive. Double-click one of the action labels to edit the labels for the move.
A simple game of poker

Chance

Alice

Bob

Move properties

Information set label

Number of members: 1

Belongs to player: Chance

Actions

<table>
<thead>
<tr>
<th>Label</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
</tr>
</tbody>
</table>
Continuing with the game, after Alice gets her card, she may either choose to "raise" or "fold." We create this move for Alice, and edit the action labels, the same way the chance move was created, except we drag the player icon from next to Alice's name to the node where we want to create the move.
C:1

1/2

Chance

High

(μ)

Low

(μ)

Alice

Bob
A simple game of poker

Chance

Alice

Bob

C: 1

High

1/2

1: 1

Low

1/2

1

2

(μ)

(μ)

(μ)

(μ)
Chance

Alice

Bob

C:1

High 1/2

Low 1/2

1:1

1

2

Move properties

Information set label

Number of members: 1

Belongs to player 1: Alice

Actions

<table>
<thead>
<tr>
<th>Label</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rais</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
If Alice folds, the game ends. However, if she raises, then Bob will have a move. He may choose to either "meet" or "pass." In either case, the game ends after Bob's move.

We add Bob's move, and set his action labels, the same way as Alice's move was created.
A simple game of poker

Chance

Alice

Bob

Move properties

Information set label

Number of members: 1

Belongs to player: 2: Bob

Actions

<table>
<thead>
<tr>
<th>Label</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
A simple game of poker (unsaved changes)

Chance

Alice

Bob

High 1/2

C:1

Low 1/2

Raise

2:1

1

2

(μ)

(μ)

(μ)

Move properties

Information set label

Number of members: 1

Belongs to player 2: Bob

Actions

Label

1 Meet

2 Pass

Cancel OK
So far, we have only drawn the interaction when Alice draws a high card. The same rules apply when she draws a low card as well.

We can easily build this by copying the tree that follows the draw of a high card to the node after the draw of a low card. To do this, hold down the control key on the keyboard, and drag from Alice's move node to the node following chance's "low" action.
Notice that when nodes are copied, they remain in the same information set as their original counterparts. Information sets are indicated by the dotted lines connecting Alice's and Bob's moves, respectively.

In our game, Alice does observe the card she draws, but Bob does not. So Alice's two moves should be in separate information sets. An easy way to accomplish this is to "reveal" the outcome of the chance move to Alice. Here's how to do that.
Right-clicking in the tree window presents a pop-up menu of editing operations.
Chance

Alice
Bob

High
1/2

raise
2:1
Fold

Meet
Pass

Raise
2:1
Fold

Meet
Pass
Now our tree is completed, and accurately reflects the moves and information structure of the game.

Now let's turn to payoffs. We suppose that at the beginning of the game, each player places a dollar in the pot. The actions "raise" and "meet" correspond to the player choosing them placing another dollar in the pot. The actions "fold" and "pass" correspond to the choosing player forfeiting the pot to the other player.

Finally, if Alice raises and Bob meets, the result depends on Alice's card. If she has a high card, she wins and takes the pot; if she has a low card, Bob wins and takes the pot.
The light grey "(u)" symbols to the right of each node represent placeholders for payoffs ("u" stands for "utility"). These are treated as a payoff of zero for all players. To create an outcome with nonzero payoffs, double-click the "(u)" symbol next to the node.

We start with the outcome in which Alice gets a high card, raises, and Bob meets her. In this case, both players have contributed two dollars to the pot (one from the ante, one from their actions). Since Alice has the high card, she wins and takes the pot, thus, her net payoff is a gain of two dollars, while Bob has lost two dollars.
A simple game of poker (unsaved changes)

Chance:
- Alice
- Bob

Alice:
- High: 1/2
- Fold: 1:1

Bob:
- Meet
- Pass

Outcome payoffs:
<table>
<thead>
<tr>
<th>Label</th>
<th>Outcome 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Choose:
- Cancel
- OK
A simple game of poker

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Outcome 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Actions:
- High
- Fold
- Raise
- Meet
- Pass
<table>
<thead>
<tr>
<th>Label</th>
<th>Alice wins big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>2</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

The diagram represents a game of poker with Alice and Bob as players. The tree diagram shows the decision points for Alice and Bob, with outcomes such as 'Meet', 'Pass', and 'Raise'. The payoff table indicates Alice's winnings for various outcomes.
Chance

Alice

Bob

High

1/2

1:1

Fold

Raise

2:1

Meet

Pass

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Alice wins big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>2</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Cancel  OK
Now we handle the case when Alice has the high card, raises, but Bob passes. Here, since Bob passes, he forfeits the dollar he put in the pot, but suffers no further loss.
A simple game of poker (unsaved changes)

Chance

Alice

Bob

High 1/2 1:1

Raise

FOLD

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Ali</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Meet 2 -2

Pass
A simple game of poker

Chance

Alice

Bob

High

1/2

1:1

Fold

Raise

2:1

Meet

Pass

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Cancel  OK
A simple game of poker

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Alice wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Actions:
- High (1/2)
- Fold (2:1)
- Raise (1:1)
- Meet
- Pass
Chance

Alice

Bob

High

1/2

1:1

Raise

Fold

C:1

2:1

Meet

Pass

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Alice wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Cancel  OK
A simple game of poker

Chance

Alice

Bob

outcome payoffs:

<table>
<thead>
<tr>
<th>Label</th>
<th>Alice wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>-1</td>
</tr>
</tbody>
</table>

Decision points:

- High: 1/2 chance of winning 1, 1/2 chance of losing 1
- Raise: 2:1 chance of winning 2, 1:1 chance of losing 1
- Fold: C:1 chance of winning 1

Actions:

- Meet
- Pass
Gambit - A simple game of poker (unsaved changes)

Chance
Alice
Bob

1/2
High

1:1

Raise

2:1

Pass

Meet

C:1

1

Outcome payoffs

Label
Alice
Bob

Alice wins

1

-1

Cancel
Similarly, when Alice folds, she forfeits her dollar to Bob.
A simple game of poker (unsaved changes)

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Outcome 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>
Chance

Alice

Bob

High: 1/2

1:1

Raise:

2:1

Meet: 2 - 2

Pass: 1 - 1

Fold:

C:1

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Bob wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Chance Graph:
- Alice decides between High and Fold.
- Bob chooses between Meet and Pass.
- Payoffs: Alice loses 2 when Meet, 1 when Pass; Bob wins 0 in both scenarios.
A simple game of poker is shown in the Gambit software. The game involves two players, Alice and Bob, with the following outcomes:

- **Chance**
  - **High** 1/2
  - **C:1** 0
  - **1:1** 0
  - **Fold** 0

- **Action**
  - **Raise** 2:1
  - **Meet** 2 -2
  - **Pass** 1 -1

The outcome payoffs are as follows:

- **Label**: Bob wins
- **Alice**: 0
- **Bob**: 0

The interface allows for the selection of outcomes and the input of payoffs for each action.
A simple game of poker

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Bob wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>-1</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>
A simple game of poker

Chance
- Alice
- Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Payoff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>-1</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

Payoffs:
- Meet: 2, -2
- Pass: 1, -1
- Fold: 1, -1
- Raise: 2, 1

Probabilities:
- High: 1/2
- C: 1

Bob wins
And, finally, if Alice gets a low card and raises, and Bob meets her, Bob wins, giving him a gain of two dollars for the game, and a loss of two dollars for Alice.
A simple game of poker

Chance

Alice

Bob

High
-1 1

Low
1:2

C:1

1:1

2:1

Raise

Meet

Pass

Fold

Meet

Pass

Fold
A simple game of poker (unsaved changes)

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Bob wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>

- High 1/2: 1:1
- Raise: 2:1
- Fold: -1 1
- Meet: 2 -2
- Pass: 1 -1

C:1
A simple game of poker

Chance

Alice

Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Bob wins bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
</tbody>
</table>
A simple game of poker (unsaved changes)

Chance
- Alice
- Bob

Outcome payoffs

<table>
<thead>
<tr>
<th>Label</th>
<th>Bob wins big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>2</td>
</tr>
<tr>
<td>Bob</td>
<td>2</td>
</tr>
</tbody>
</table>

- High
  - 1/2
  - 1:1
  - Raise
    - 2:1
      - Meet: 2, -2
      - Pass: 1, -1
    - Fold: -1, 1
- C: 1
Recall that, unless Alice raises and Bob meets, the value of the card Alice holds doesn’t affect the payoffs. So, in the remaining two cases, we can copy the payoffs from nodes where we already created them.

Payoffs can be copied by holding down the control key on the keyboard while dragging from the payoffs to be copied to the node where the payoffs should also be assigned. We now do this for both the remaining cases.
A simple game of poker

Chance

Alice

Bob

- High: 1/2
- Low: 1/2

C:1

1:1

Raise: 2:1

Meet: 2 -2
Pass: -1 1

Fold: -1 1

Raise: 2:1

Meet: -2 2
Pass: 1 -1

Fold: -2 2
This completes our specification of the game.

A next step in the analysis is to consider whether any of the actions available to the players are dominated. Gambit provides a toolbar to facilitate this analysis, which can be toggle on or off using the Tools menu.
Let us see whether any actions are dominated, either strictly or weakly, in this game.
Gambit - A simple game of poker (unsaved changes)

Hide actions which are strictly dominated:

All actions shown

Show only reachable nodes

Chance

Alice

Bob

Chance

C:1 (1/2)

High: 1:1 (1/2)

Raise: 2:1 (ω)

Meet: 2 -2

Pass: 1 -1

Bob

Low: 1:2 (ω)

Raise: 2:1 (ω)

Meet: -2 2

Pass: 1 -1

Fold

Fold: -1 1
The arrow buttons step through the levels of elimination of dominated actions. The arrow button pointing rightward performs one round of elimination.
We see that it is dominated for Alice to fold when she has a high card. This action is now eliminated from consideration, indicated by its absence from the game tree.

To see if further actions can be eliminated, we click the right-arrow button again.
No further actions are removed from the tree; thus, we have reached the end of the elimination process. The right arrow button is now disabled on the toolbar to indicate this.

The left arrow button, similarly, would undo one step of the elimination process. The other arrow buttons jump to the beginning and the end of the process. We now restore the game in full on the screen by jumping to the beginning.
Next, we might be interested in the reduced strategic form of the game.

Gambit computes this automatically. We can toggle between the extensive and strategic forms by using the table icon on the toolbar.
A simple game of poker.
Gambit uses a shorthand to summarize strategies. For Alice, each strategy lists the action she takes at each of her information sets. Thus, this strategy corresponds to her taking her second action (fold) at her first information set (when she has the high card), and her taking her first action (raise) at her second information set (when she has the low card).
Let's repeat the dominance analysis on the strategic form. The dominance toolbar works analogously on the strategic form as it did in the extensive form.
<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>(\frac{1}{2})</td>
<td>(-\frac{1}{2})</td>
</tr>
<tr>
<td>21</td>
<td>(-\frac{3}{2})</td>
<td>(\frac{3}{2})</td>
</tr>
<tr>
<td>22</td>
<td>(-1)</td>
<td>1</td>
</tr>
<tr>
<td>Alice</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The game matrix represents a simple game of poker with strategies for Alice and Bob. The strategies are represented by numbers, and the payoffs are shown in the table. Alice has strategies 11 and 12, while Bob has strategy 1.
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>12</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>21</td>
<td>1.5</td>
<td>4.5</td>
</tr>
<tr>
<td>22</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Alice</td>
<td>Bob</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td></td>
</tr>
</tbody>
</table>
| 1
<p>| 11   | 1   | 0.0 |
| 12   | 0.5 | 0.5 |
| 21   | -1  | 1   |
| 22   | 1   | 1   |</p>
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Alice</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

Observe that Alice's strategies "21" and "22" have been eliminated. Recall that these both involve Alice playing her second action, fold, at her first information set, when she has the high card. This is the action we earlier determined to be dominated.
### Alice vs. Bob: A Simple Game of Poker

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>11</strong></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>12</strong></td>
<td>½</td>
<td>½</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Next round of elimination
<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>Next round of elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
No more strategies have been removed, so we have come to the end of the elimination process. We will now hide the dominance toolbar.

It should be noted that this elimination toolbar is entirely for the benefit of the analyst. Gambit automatically takes advantage of dominance information where appropriate in computing equilibria.
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>Bob</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Speaking of which, what are the Nash equilibria of this game? The algorithms for finding these are available by clicking on the calculator button on the toolbar.
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>-1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>21</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>Bob</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Compute Nash equilibria of this game
There are many methods for finding Nash equilibria, each with their own strengths and weaknesses. Furthermore, for some games special methods are available that are more efficient.

To get started, though, you don't have to worry about this, since Gambit provides "recommended" methods which typically work well. Simply select the number of equilibria you want to find. Here, let's find all of them.
<table>
<thead>
<tr>
<th>Alice</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bob

Compute Nash equilibria
- Compute all Nash equilibria
- with Gambit's recommended method
- using the extensive game

[OK] [Cancel]
<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1/2</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
</tr>
</tbody>
</table>

The payoff matrix for the game is as follows:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>-1/2</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>-3/2</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>-3/2</td>
<td>3</td>
</tr>
</tbody>
</table>

To compute the Nash equilibria, choose one of the following options:

- Compute all Nash equilibria
- With Gambit's recommended method
- Using the extensive game
Almost immediately, the computation completes, and presents us with one strategy profile, which is the unique Nash equilibrium of this game.
Computing Nash equilibria

The computation has completed.
Number of equilibria found so far: 1

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0</td>
<td>0</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
### Computing Nash equilibria

The computation has completed. Number of equilibria found so far: 1

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0</td>
<td>0</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
Notice that the player panel has updated to indicate the payoffs the players receive in this equilibrium. Alice can expect to gain about 33 cents per game, and Bob can expect to lose about 33 cents per game.

To interpret the strategies better, let’s look at them converted back into strategies on the extensive form. To do this, toggle off the table tool on the toolbar.
<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0</td>
<td>0</td>
<td>0.667</td>
<td>0.333</td>
</tr>
</tbody>
</table>
Notice that Gambit graphically presents the strategies on the tree for visualization. For example, the black segment on Bob's meet action is twice as long as that for his pass action, indicating he is twice as likely to meet as to pass when he gets to play.
With the addition of the list of equilibria, we can't quite see the whole game tree. We could drag the sash separating the windows to resize them. Or, we can zoom out on the game tree to fit it into the space we have. We'll do the latter here.
Chance

Alice
Payoff: 0.3333

Bob
Payoff: -0.3333

1: Raise 1: Fold 2: Raise 2: Fold 1: Meet 1: Pass
0.3333 0.6667 0.6667 0.3333

C:1
Low

1: Raise 1: Fold 2: Raise 2: Fold 1: Meet 1: Pass
0.3333 0.6667 0.6667 0.3333

Set magnification to see entire tree
In the equilibrium, Alice always raises when she has the high card. (This makes sense; after all, folding is dominated!)
When Alice has the low card, she should raise one-third of the time. In other words, "bluffing" is in fact good strategy in this game!
Finally, we see that Bob should meet two-thirds of Alice's raises.
We can navigate around the game tree to get a better idea of how the play of the game proceeds. Click on any node to get more information.
We have clicked on the node where Alice moves after getting a high card. We see, for example, that when she draws a high card, she expects to win about 1.67 in that play of the game (her "infoset value").

More interesting is Bob's move, since he has imperfect information in this game. A handy shortcut, especially in larger trees, is to click on a column heading in the profile list. This selects the first node in the corresponding information set.
At the top node in Bob’s information set, Bob places a belief of .75. That is to say, conditional on Alice raising, Bob believes there is a three-fourths chance he is actually at the top node in his information set, or, in other words, a three-fourths chance that Alice has the high card.
Once we have the game built, we can play around with parameters to see how strategies change. For example, what if we change the rules so that Alice now has a two-thirds chance of drawing the high card?
A simple game of poker (unsaved changes)

**Chance**

**Alice**
Payoff: 0.3333
Node value: -1.0000

**Bob**
Payoff: -0.3333
Node value: 1.0000
Node reached: 0.1667
Infoset value: -1.0000
Infoset reached: 0.6667
Belief: 0.2500

---

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: Raise</th>
<th>1: Fold</th>
<th>2: Raise</th>
<th>2: Fold</th>
<th>1: Meet</th>
<th>1: Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
Chance

Alice
Payoff: 0.3333

Bob
Payoff: -0.3333
Node reached: 0.1667
Infoset value: -1.0000
Infoset reached: 0.6667
Belief: 0.2500

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: Raise</th>
<th>1: Fold</th>
<th>2: Raise</th>
<th>2: Fold</th>
<th>1: Meet</th>
<th>1: Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0</td>
<td>0.3333</td>
<td>0.6667</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
Gambit - A simple game of poker (unsaved changes)

Chance

Alice
Payoff: 0.3333
Node value: 0.3333

Bob
Payoff: -0.3333
Node value: -0.3333

Move properties

Information set label: [blank]
Number of members: 1
Belongs to player: Chance

Actions

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: Raise</th>
<th>1: Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Label</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Cancel | OK
Alice
Payoff: 0.3333
Node value: 0.3333

Bob
Payoff: -0.3333
Node value: -0.3333

Chance

Move properties
Information set label: 
Number of members: 1
Belongs to player: Chance

Actions
<table>
<thead>
<tr>
<th></th>
<th>Label</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>2/3</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Alice
Payoff: 0.3333
Node value: 0.3333

Bob
Payoff: -0.3333
Node value: -0.3333

Gambit - A simple game of poker (unsaved changes)

Move properties
Information set label:
Number of members: 1
Belongs to player: Chance

<table>
<thead>
<tr>
<th>Actions</th>
<th>Label</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>$\frac{2}{3}$</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>$\frac{1}{3}$</td>
</tr>
</tbody>
</table>
Now we again send Gambit looking for equilibria.
A simple game of poker

Chance

Alice

Bob

Liap Value | 1: Raise | 1: Fold | 2: Raise | 2: Fold | 1: Meet | 1: Pass
--- | --- | --- | --- | --- | --- | ---
--- | --- | --- | --- | --- | --- | ---
--- | --- | --- | --- | --- | --- | ---
### Computing Nash equilibria

The computation has completed. Number of equilibria found so far: 1

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.6667</td>
<td>0.3333</td>
<td>0</td>
<td>0</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
The computation has completed.
Number of equilibria found so far: 1

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.6667</td>
<td>0.3333</td>
<td>0</td>
<td>0</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
Computing Nash equilibria

The computation has completed. Number of equilibria found so far: 1

<table>
<thead>
<tr>
<th>#</th>
<th>Liap Value</th>
<th>1: 11</th>
<th>1: 12</th>
<th>1: 21</th>
<th>1: 22</th>
<th>2: 1</th>
<th>2: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.6667</td>
<td>0.3333</td>
<td>0</td>
<td>0</td>
<td>0.6667</td>
<td>0.3333</td>
</tr>
</tbody>
</table>
Observe that Alice's strategy has changed. When she has the low card, her bluffing probability increases to two-thirds, up from the one-third we computed in the earlier version.
Bob's belief at his top node remains three-quarters. This quantity is crucial in this game, since it is this probability of Alice having a high card that exactly makes Bob indifferent between meeting and passing.

The game theorist's advice to Alice in this game is to play in such a way that Bob will think there is a three-fourths chance she has a high card, given that she raises. As we lower the probability of a bad hand, Alice should raise the frequency with which she bluffs to compensate, so that Bob will assess this desired probability.
This concludes this tutorial on using Gambit.

You can view or download other tutorials from the Gambit website at

http://econweb.tamu.edu/gambit

Happy gaming!