

# istgame.sty

## Drawing Game Trees with TikZ

In-Sung Cho  
 ischo <at> ktug.org

2017/02/13 version v0.9

### Abstract

This package provides macros based on TikZ to draw a game tree. The main idea underlying the core macros here is the completion of a whole tree using a sequence of simple ‘parent-child’ tree structures, with no longer nested relations involved like down to grandchildren or to great-grandchildren.

KEYWORDS: game trees, tikz, nodes, branches, information sets, subgames

## Table of Contents

0	Changes	1
1	Getting Started	1
1.1	getting-started examples	1
1.2	controlling the length and the direction of branches	2
2	Macros and Syntax	3
2.1	environment <code>istgame</code> and node styles	3
2.2	core macros: <code>\istroot</code> and <code>\istb</code>	4
2.3	controlling the positions of payoffs, owners, and action labels	7
2.4	information sets and some supplement macros	11
2.5	continuum of branches: <code>\istcntm</code>	13
3	Game Tree Examples	15
3.1	simple examples	15
3.2	larger game trees	16
3.3	some more extensive games	17
3.3.1	centipede game	17
3.3.2	poker game	18
3.3.3	poker game: growing to the right	19
3.3.4	signaling game	20
3.4	subgames and backward induction	21
	Version history	22
	Acknowledgement	23
	References	23

## 0 Changes

A considerable number of macro names have been changed in the version 0.8 (Jan. 17, 2017) of this package. The following old macro names in any previously written documents using codes in `istgame` ver. 0.7 or before, should be replaced by the new names, accordingly.

version 0.7 or before	ver 0.8 or later
<code>\xdistance</code>	<code>\xtdistance</code>
<code>\xDot</code>	<code>\xtNode</code>
<code>\xInfoset</code>	<code>\xtInfoset</code>
<code>\xInfoset*</code>	<code>\xtInfoset*</code>
<code>\xInfosetOwner</code>	<code>\xtInfosetOwner</code>
<code>\xActionLabel</code>	<code>\xtActionLabel</code>
<code>\xPayoff</code>	<code>\xtPayoff</code>
<code>\ShowTerminalNodes</code>	<code>\xtShowTerminalNodes</code>
<code>\HideTerminalNodes</code>	<code>\xtHideTerminalNodes</code>
<code>\levdist</code>	<code>\xtlevdist</code>
<code>\sibdist</code>	<code>\xtsibdist</code>

Here, the prefix ‘xt’ stands for ‘extensive tree,’ so `\xtdiatnace` can be read as ‘extensive tree distance.’

## 1 Getting Started

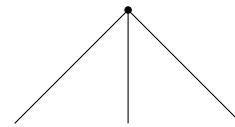
The package `istgame` provides macros to draw game trees. The core macros provided with this package are `\istroot`, `\istb`, and `\endist`. `\istroot` pins down the root of a tree or a subtree, `\istb` represents a branch, and `\endist` indicates the end of drawing a simple tree. Without `\endist`, the tree is not actually drawn.

Here, the prefix ‘ist’ stands for ‘insung’s simple tree.’ Note that the package `istgame` depends on the packages `tikz`, `xparse`, and `expl3`.

### 1.1 getting-started examples

Let us get started with a simple self-explanatory example:

```
%% \usepackage{istgame}
\begin{istgame}
\istroot(0)(0,0) % names the root as (0) at (0,0)
  \istb % endpoint will be (0-1), automatically
  \istb % endpoint will be (0-2)
  \istb % endpoint will be (0-3)
\endist % end of simple (parent-child) structure
\end{istgame}
```



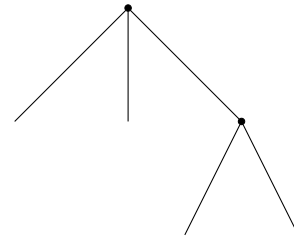
The resulting tree has the height of 15mm and the distance between two neighbor endpoints is also 15mm, in their defaults. In `TikZ`, the height is called the **level distance** and the distance between the endpoints the **sibling distance**.

Basically, to draw a whole game tree, we just repeat the simple `\istroot`-`\istb`-`\endist` sequence. If the second parenthesis argument of `\istroot` is omitted, it is regarded as (0,0) in default, otherwise it is necessary to specify the coordinate from which a subtree starts.

```

\begin{istgame}
\istroot(0) % names the root as (0) at (0,0)
  \istb % endpoint will be (0-1), automatically
  \istb % endpoint will be (0-2)
  \istb % endpoint will be (0-3)
\endist % end of simple (parent-child) structure
\istroot(c)(0-3) % names the subroot as (c) at (0-3)
  \istb % endpoint will be (c-1)
  \istb % endpoint will be (c-2)
\endist
\end{istgame}

```



## 1.2 controlling the length and the direction of branches

The length and the direction of a branch can be controlled by the command `\xtdistance`.

```

\def\xtlevdist{15mm} \def\xtsibdist{15mm} %% (default) initial values for
distances
\NewDocumentCommand\xtdistance{0{1}mG{15mm}}
{
  \renewcommand\xtlevdist{#2}
  \renewcommand\xtsibdist{#3}
  \tikzstyle{level #1}= [level distance=\xtlevdist,sibling distance=\xtsibdist]
}

```

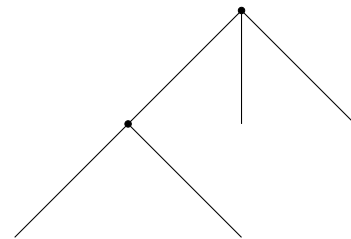
- `\def\xtlevdist{15mm}` and `\def\xtsibdist{15mm}` set the default values for the level distance and the sibling distance.
- `\xtdistance` sets or resets at any time needed the level distance and the sibling distance with an option for level depth (default=1).
- syntax: `\xtdistance[<level depth num>]{<level dist num>}{<sibling dist num>}`

With the current package `istgame`, the level depth of 1 suffices. (For more details about the level depth, see TikZ manual or Chen(2013).)

```

\begin{istgame}
\xtdistance{15mm}{15mm} % default
\istroot(0) % names the root (0) at (0,0)
  \istb % endpoint will be (0-1), automatically
  \istb % endpoint will be (0-2)
  \istb % endpoint will be (0-3)
\endist % end of simple (parent-child) structure
\xtdistance{15mm}{30mm} % controls distances
\istroot(a)(0-1) % names the subroot as (a) at (0-3)
  \istb % endpoint will be (a-1)
  \istb % endpoint will be (a-2)
\endist
\istroot(a)
\end{istgame}

```



In fact, the core macros are much more powerful. `\istroot` controls the direction that a tree grows to, node styles, the node owner and its position, the height and sibling distance of a current simple tree, etc. `\istb` specifies the growing direction of a branch, branch line styles, branch color, the label of action, and the payoffs and their position. The next section describes the core macros and others in more details.

## 2 Macros and Syntax

### 2.1 environment `istgame` and node styles

#### environment `istgame`

This package provides `istgame` environment, which is simply `tikzpicture` environment so accepts all the options of `tikzpicture` environment. The default font size is set as `font=\normalsize`. To change the default font size of `istgame` environment, for example, to `scriptsize` you can do `\setistgamefontsize{\scriptsize}`. You can also locally change the font size by using the `font` option key, like `\begin{istgame}[font=\scriptsize]...`

```
\def\istgamefontsize{\normalsize}
\NewDocumentCommand\setistgamefontsize {m}
  {\renewcommand*\istgamefontsize{#1}
  }

\NewDocumentEnvironment{istgame}{0{}} % accepts tikzpicture options
  {\begin{tikzpicture}[%
    edge from parent path=%
      {(\tikzparentnode\istparentanchor) -- (\tikzchildnode\istchildanchor)},
    font=\istgamefontsize,#1
  ]
  }
  {\end{tikzpicture}
  }
```

#### node styles:

`tikzstyle`'s of the six basic node types are predefined.

- zero node : draws nothing
- null node :  $\cdot$  (very small node, not expected to be used)
- solid node :  $\bullet$  (default node style)
- hollow node :  $\circ$
- rectangle node :  $\square$
- ellipse node :  $\bigcirc$

For some special cases, you may want to change the node styles of the basic nodes, including the minimum size as mandatory argument. This can be done by `\setist<...>NodeStyle`, except the zero node.

#### syntax:

```
\setistNullNodeStyle[<color>]{<min-size dim>}
\setistSolidNodeStyle[<color>]{<min-size dim>}
\setistHollowNodeStyle[<color>]{<min-size dim>}[<bg color>][<opacity num>]
\setistRectangleNodeStyle[<color>]{<min-size dim>}[<bg color>][<opacity num>]
\setistEllipseNodeStyle[<color>]{<min-size dim>}[<bg color>][<opacity num>]
```

```
% Examples:
\begin{istgame}\setistSolidNodeStyle[blue]{10pt}
  \istroot(0)[solid node]\endist\end{istgame}\[1ex]
\begin{istgame}\setistHollowNodeStyle[blue]{10pt}[yellow]
  \istroot(0)[hollow node]\endist\end{istgame}\[1ex]
\begin{istgame}\setistRectangleNodeStyle{10pt}[red][.5]
  \istroot(0)[rectangle node]\endist\end{istgame}\[1ex]
\begin{istgame}\setistEllipseNodeStyle[blue]{10pt}[green]
  \istroot(0)[ellipse node]\endist\end{istgame}\[1ex]
\begin{istgame}\setistNullNodeStyle[blue!20]{10pt}
  \istroot(0)[null node]\endist\end{istgame}
```



These basic node styles have their aliases, for convenience, for who are familiar with the game-theoretic terminology.

```
\tikzstyle{decision node}=[solid node] % decision nodes
\tikzstyle{terminal node}=[solid node] % terminal nodes
\tikzstyle{initial node}=[hollow node]
\tikzstyle{chance node}=[hollow node]
```

The set of all nodes of a game tree can be partitioned into the set of **decision nodes** and that of **terminal nodes**.

You can use **initial node** or **chance node** to distinguish the root (or the initial node) of a game tree from decision nodes. You can also use **chance node** to express a chance node of a game tree.

You can also change the node styles, like `\setistDecisionNodeStyle[blue]{3pt}`. Additional convenient node aliases are also provided.

```
\tikzstyle{box node}=[rectangle node]
\tikzstyle{square node}=[rectangle node]
\tikzstyle{oval node}=[ellipse node]
```

You can change the node styles, like `\setistBoxNodeStyle[blue]{3pt}[green][.5]`.

## 2.2 core macros: `\istroot` and `\istb`

### `\istroot`

The macro `\istroot` defines the root of a game or a subgame, specifies the owner of the node, and does some more.

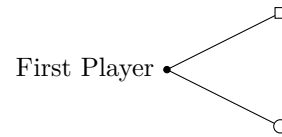
```
% \istroot
syntax: \istroot[<grow key>](<coor>)(<coor>)[<node style>]%
      <owner label angle>{<owner>}+<lev-distance>..<sib-distance>+
defaults: [south]() (0,0)[decision node]<above>{}+15mm..15mm+
arguments: 1 mandatory, 7 optional arguments
  [grow] % the direction of growing <default: south>
  (coor) % node name: mandatory
  (coor) % node is at (coor) <default: (0,0)>
  [node style] % node style <default: decision node>
  <angle> % position of owner name <default: above>
  {owner} % owner name
  +level dist..sibling dist+ % <defaults: 15mm,15mm>
```

Examples:

```

\begin{istgame}
\istroot[right](0)<180>\{First Player\}
\istb \istb \endist
\xtNode(0-1)[oval node] \xtNode(0-2)[square node]
\end{istgame}

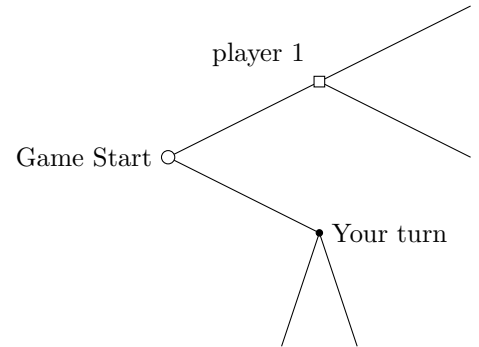
```



```

\begin{istgame}
\xtdistance{20mm}{20mm}
\istroot[right](0)[oval node]<left>\{Game Start\}
\istb \istb \endist
\istroot(a)(0-1)<right>\{Your turn\}+15mm..10mm+
\istb \istb \endist
\istroot[right](b)(0-2)[box node]<135>\{player 1\}
\istb \istb \endist
\end{istgame}

```



remark: `\def\istgrowdirection{<grow key>}`  
`<grow key>` assigns its value to `\istgrowdirection`.

The direction of tree growing can be specified by **degrees** or the key values such as **left**, **right**, **south**, **down**, **up**, and so forth. One complication about this is that `\istgrowdirection` is internally used in the definition of `\istb` to control the label position for payoffs. However, for the label position **below** and **above** are good, but not **down** and **up**. So using **down** or **up** to specify the tree growing direction is discouraged.

The last two options of `\istroot` specify the **level distance** and the **sibling distance**. This change of distances is valid only for a corresponding simple tree, while distance change by `\xtdistance` is valid within the current `istgame` environment unless it is changed again by `\xtdistance`. Do not forget, when you use decimal distances, to delimit with braces the decimal dimensions, like `+{10.5mm}..{10.2mm}++`.

### `\istroot*`

The starred version `\istroot*` allows us to draw a bubble (in default, using `oval node`) with a node owner (or a game player) in it.

```

% \istroot*
syntax: \istroot* [<grow key>] (<coor>) (<coor>) [<node style>] {\<player>} +levd..sibd+
default: [south] () (0,0) [oval node] {} +15mm..15mm+

```

Examples: (codes are the same as above but with stars)

```

\begin{tikzpicture}
\istroot*[right](0)<180>\{First Player\}
\istb \istb \endist
\xtNode(0-1)[oval node] \xtNode(0-2)[square node]
\end{tikzpicture}

```

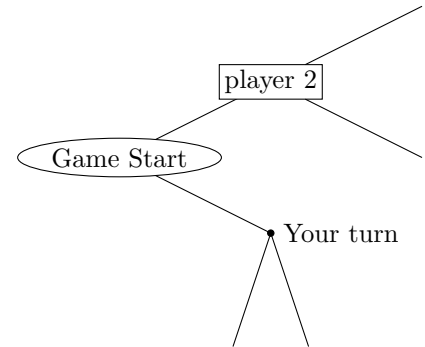


Notice that `istgame` environment can be replaced by `tikzpicture` environment.

```

\begin{istgame}
\xtdistance{20mm}{20mm}
\istroot*[right](0)[oval node]<left>{Game Start}
\istb \istb \endist
\istroot*(a)(0-1)<right>{Your turn}+15mm..10mm+
\istb \istb \endist
\istroot*[right](b)(0-2)[box node]<135>{player 2}
\istb \istb \endist
\end{istgame}

```



Notice that the angle `<180>`, `<left>`, or `<135>` specifying the position of the owner's name is redundant with `\istroot*`.

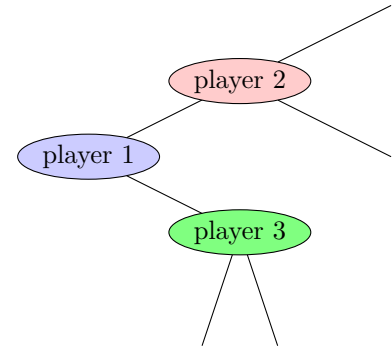
What if you want to paint some color into the background of each oval node? You can do this simply by adding `tikz node style` options.

Example:

```

\begin{istgame}
\xtdistance{20mm}{20mm}
\istroot*[right](0)[oval node,fill=blue!20]{player 1}
\istb \istb \endist
\istroot*(a)(0-1)[oval node,fill=green!50]{player 3}%
+15mm..10mm+
\istb \istb \endist
\istroot*[right](b)(0-2)[oval
node,fill=red!20]{player 2}
\istb \istb \endist
\end{istgame}

```



```

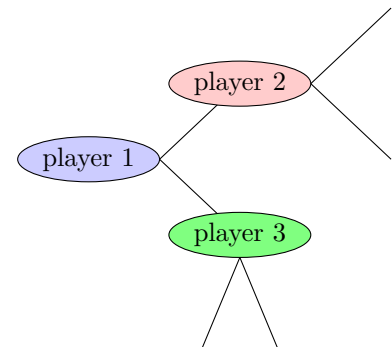
% syntax: \setistnodeanchors, \setistparentanchor
\setistnodeanchors{<parent anchor>}{<child anchor>}
% defaults:
{}{north}

```

```

% Example: \setistnodeanchors
\begin{istgame}
\setistnodeanchors{east}{west}
\xtdistance{20mm}{20mm}
\istroot*[right](0)[oval node,fill=blue!20]{player 1}
\istb \istb \endist
\setistparentanchor{south}
\istroot*(a)(0-1)[oval node,fill=green!50]{player 3}%
+15mm..10mm+
\istb \istb \endist
\setistnodeanchors{east}{west}
\istroot*[right](b)(0-2)[oval
node,fill=red!20]{player 2}
\istb \istb \endist
\end{istgame}

```



## `\istb`

The macro `\istb` draws a branch with an optional action label and payoffs. Action labels and payoffs are defined to be in math mode, so you can use `\text{...}` when you want to typeset them in text mode.

```
% \istb
syntax: \istb<grow, distance>[<line style>]{<action>}[<pos>]{<payoff>}[<pos>]
default: [south]{-}{ }[ ]{\istgrowdirection}
```

remark: What is `\istgrowdirection` and what for?

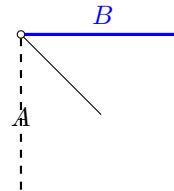
`\istgrowdirection` specifies the position of payoffs with `south` as default.  
`\istgrowdirection` is predefined and has the value of `<grow key>` typed in `\istroot`.

If a tree grows to the right, `\istgrowdirection` makes the payoff put at the right of corresponding terminal node. We will see in more details below when we discuss how to put payoffs.

Besides the direction of a branch, `\istb` also controls the line style of a branch, the corresponding action label, and the payoffs and their position. To specify the position for the action label or payoffs within `\istb` or `\istb*`, you can use the letter `l` as an abbreviation for `left`, `r` for `right`, `a` for `above`, and `b` for `below`.

Example:

```
\begin{istgame}
\istroot[-45](0)[initial node]
\istb[dashed,thick]{A}
\istb
\istb[blue,very thick]{B}[above]
\endist
\end{istgame}
```

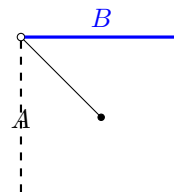


`\istb*`

The starred version `\istb*` prints a “terminal node” at the end of the corresponding branch.

Example:

```
\begin{istgame}
\istroot[-45](0)[initial node]
\istb[dashed,thick]{A}
\istb*
\istb*[blue,very thick]{B}[above]
\endist
\end{istgame}
```



Each terminal node is printed by each execution of the command `\istb*`. All the terminal nodes can be printed by one time execution of `\xtShowTerminalNodes`. You can change the style of terminal nodes by specifying it as an optional argument, like `\xtShowTerminalNodes[box node]`. Though `\xtHideTerminalNodes` can turn off the effect of `\xtShowTerminalNodes`, it is safe to use `\xtShowTerminalNodes` and `\xtHideTerminalNodes` in an `istgame` environment in order to avoid to get unexpected results. Notice also that `\istb*` overrides all the effect of these two macros by printing a solid node.

## 2.3 controlling the positions of payoffs, owners, and action labels

### how to put payoffs

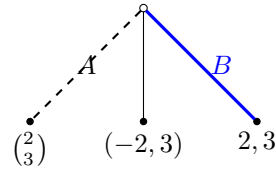
The direction of where payoffs are put from a terminal node follows `\istgrowdirection` typed in as the first optional argument of `\istroot`. The default direction is `south` and can be changed by `\setistgrowdirection`. `\setistgrowdirection[north]` changes the default direction to `north`.



```

\begin{istgame}
\xtShowTerminalNodes
\istroot(0)[initial node]
\istb[dashed,thick]{A}{\binom{2}{3}}
\istb{}{(-2,3)}
\istb[blue,very thick]{B}[right]{2,3}
\endist
\end{istgame}

```

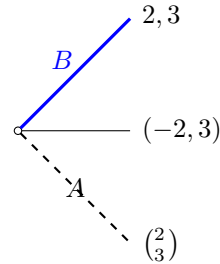


`grow=south` in default, so `\istgrowdirection=south` (or below).  
This result is showing payoffs below (or at the south of) the terminal node.

```

\begin{istgame}
\istroot[right](0)[initial node]
\istb[dashed,thick]{A}{\binom{2}{3}}
\istb{}{(-2,3)}
\istb[blue,very thick]{B}[shift=(135:7pt)]{2,3}
\endist
\end{istgame}

```

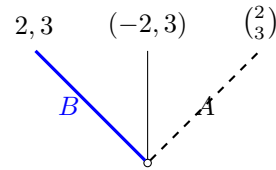


`grow=right=\istgrowdirection`, so the payoffs are put on the right.

```

\begin{istgame}
\setistgrowdirection{north}
\istroot(0)[initial node]
\istb[dashed,thick]{A}{\binom{2}{3}}
\istb{}{(-2,3)}
\istb[blue,very thick]{B}[left]{2,3}
\endist
\end{istgame}

```

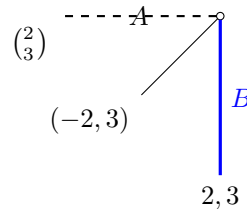


`grow=north=\istgrowdirection`, so the payoffs are put above the terminal nodes.

```

\begin{istgame}
\istroot[south west](0)[initial node]
\istb[dashed,thick]{A}{\binom{2}{3}}
\istb{}{(-2,3)}
\istb[blue,very thick]{B}[right]{2,3}[b]
\endist
\end{istgame}

```



`grow=south west=\istgrowdirection`, so the payoffs are put below left of the terminal node.

You can adjust the direction of any payoff to put by specifying an option right after the payoff, like `\istb[blue,very thick]{B}[right]{2,3}[below]`. You can use the abbreviations `[l]`, `[r]`, `[a]`, and `[b]` for `[left]`, `[right]`, `[above]`, and `[below]`, respectively. The abbreviations `[al]`, `[ar]`, `[bl]`, and `[br]` can be used for `[above left]`, `[above right]`, `[below left]`, and `[below right]`, respectively, to put payoffs. Notice also that, instead of the directional words, you can use degrees, like `\istb[blue,very thick]{B}[right]{2,3}[-90]`.

### manipulating the position of action labels

In order to determine the direction of action labels for branches to put, you can use degrees and directional words and their abbreviations such as mentioned above. (Internally, the abbreviations for payoffs and that those for action labels work slightly differently in terms of `xshift` and `yshift`.)

```

\begin{istgame}[scale=1.2]
\xtShowTerminalNodes
\xtdistance{12mm}{16mm}
\istroot(0)[chance node]
  \istb<grow=0>{\fbox{$a$}}[a]
  \istb<grow=90>{\fbox{$l$}}[l]
  \istb<grow=180>{\fbox{$b$}}[b]
  \istb<grow=-90>{\fbox{$r$}}[r]
\endist
\end{istgame}

```

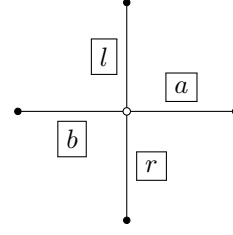


Figure 1: action labels in defaults with l, r, a, and b

When you use these abbreviations you can manipulate the horizontal and/or the vertical shifts toward branches by using `\setistactionlabelshift`.

syntax: `\setistactionlabelshift{<xshift dim> for l and r}{<yshift dim> for a and b}`  
 default: `xshift=<0pt>, yshift=<0pt>`

For example, `\setistactionlabelshift{-3pt}{5pt}` draws the labels put left and right by 3pt to the branch and push those put above and below 5pt away from the branch.

```

\begin{istgame}[scale=1.2]
\xtShowTerminalNodes
\setistactionlabelshift{-3pt}{5pt} % look here
\xtdistance{12mm}{16mm}
\istroot(0)[chance node]
  \istb<grow=0>{\fbox{$a$}}[a]
  \istb<grow=90>{\fbox{$l$}}[l]
  \istb<grow=180>{\fbox{$b$}}[b]
  \istb<grow=-90>{\fbox{$r$}}[r]
\endist
\end{istgame}

```

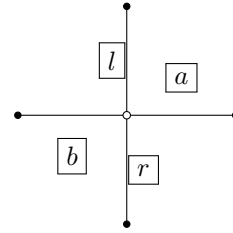
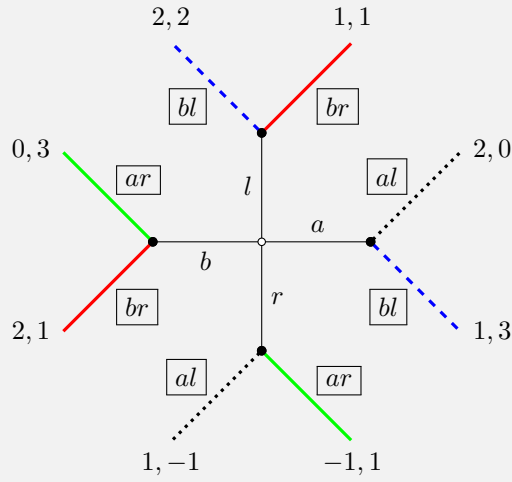


Figure 2: action labels shifted: comparison

### (experimental) efficient way of placing action labels

You can also use the abbreviations `al`, `ar`, `bl`, and `br` to represent above left, above right, below left, and below right, respectively, to position action labels. Precise representation of abbreviations is as follows:

- `[al]` represents [above left, `xshift=0.5pt, yshift=-3.5pt, black`]
- `[ar]` represents [above right, `xshift=-0.5pt, yshift=-3.5pt, black`]
- `[bl]` represents [below left, `xshift=0.5pt, yshift=3.5pt, black`]
- `[br]` represents [below right, `xshift=-0.5pt, yshift=3.5pt, black`]



```

\begin{istgame}[scale=1.2]
\def\xbox#1{\fbox{$#1$}} \def\ybox#1{#1}
\xtdistance{12mm}{16mm}
\istroot(0)[chance node]
\istb<grow=0>{\ybox{a}}[a]
\istb<grow=90>{\ybox{l}}[l]
\istb<grow=180>{\ybox{b}}[b]
\istb<grow=-90>{\ybox{r}}[r]
\endist
\begin{scope}[line width=1.2]
\xtdistance{10mm}{20mm}
\istroot[90](N)(0-2)
\istb[red]{\ebox{br}}[br]{1,1}
\istb[blue,dashed]{\ebox{bl}}[bl]{2,2}
\endist
\istroot[180](W)(0-3)
\istb[green]{\ebox{ar}}[ar]{0,3}
\istb[red]{\ebox{br}}[br]{2,1}
\endist
\istroot[-90](S)(0-4)
\istb[dotted]{\ebox{al}}[al]{1,-1}
\istb[green]{\ebox{ar}}[ar]{-1,1}
\endist
\istroot[0](E)(0-1)
\istb[blue,dashed]{\ebox{bl}}[bl]{1,3}
\istb[dotted]{\ebox{al}}[al]{2,0}
\endist
\foreach \x in {1,...,4} {\xtNode(0-\x)}
\end{scope}
\end{istgame}

```

Figure 3: positioning action labels

In Figure 3,  $\boxed{al}$ 's are put at the same position from dotted branches, and so are the other labels from their corresponding branches, like  $\boxed{bl}$ 's from blue dashed branches.

You can also use `\setistactionlabelposition` to put and push labels horizontally and vertically.

syntax: `\setistactionlabelposition{<horizontal shift dim>}{<vertical shift dim>}`  
default: `{0.5pt}{3.5pt}`

When the dimensions get bigger than the defaults the labels get closer to the corresponding branches, and when smaller the labels get farther from their branches. See how it is used in the example of the subsection 3.3.3.

## 2.4 information sets and some supplement macros

### information sets

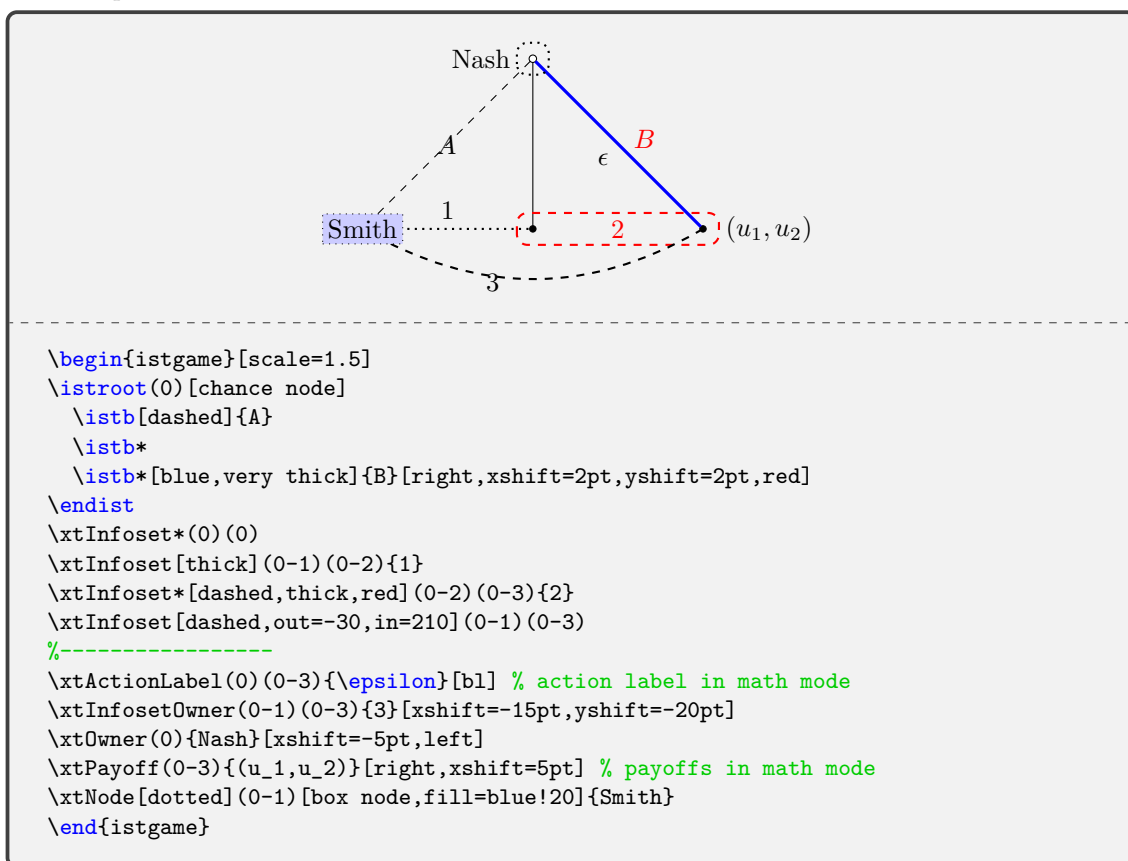
syntax: `\xtInfoset[<info line type>](<from>)(<to>){<owner>}[<pos>,<node opt>]`  
 syntax: `\xtInfoset*[<info line type>](<from>)(<to>){<owner>}[<pos>,<node opt>]`

The starred version `\xtInfoset*` prints a bubble type information set. (experimental!)

### supplement macros

syntax: `\xtInfosetOwner(<from>)(<to>){<owner>}[<pos>,<node opt>]`  
 syntax: `\xtActionLabel(<from>)(<to>){<action>}[<pos>,<node opt>]`  
 syntax: `\xtOwner(<coord>){<owner>}[<pos>,<node opt>]`  
 syntax: `\xtPayoff(<coord>){<payoff>}[<pos>,<node opt>]`  
 syntax: `\xtNode[<opt>](<coord>)[<node type>,<node opt>]{<owner>}`

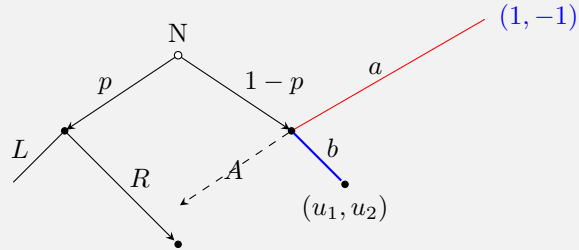
Example:



The supplement macros depend on the coordinates defined in the sequence of `\istroot`–`\istb`–`\endist` and print their corresponding objects on or around the specified coordinates. To specify the `[<pos>]` option for the above supplement macros (other than `\xtNode`), you can use the abbreviations `[l]`, `[r]`, `[a]`, `[b]`, `[al]`, `[ar]`, `[bl]`, and `[br]`. Each of these abbreviations work only when used alone without any other option keys within brackets. For example, the option `[r]` or `[right]` or `[right,xshift=5pt]` works, but not `[r,xshift=5pt]`.

## various branch types, directions, and lengths

What follows is an example of how to deal with various line types, directions, and lengths of branches.



```
\begin{istgame}[>=stealth,shorten >=.8pt]
\istroot(0)[chance node]{N}+10mm..30mm+
  \istb[->]{p}[al]
  \istb[->]{1-p}[ar]
\endist
\istroot(1)(0-1)+15mm..30mm+
  \istb<level distance=7mm,sibling distance=14mm>{L}[al]
  \istb*[->]{R}[ar]
\endist
\istroot(2)(0-2)+10mm..15mm+
  \istb[->,dashed]{A}
  \istb<grow=30,level distance=30mm>[red]{a}[al]{(1,-1)}[[blue]right]
  \istb*<grow=-45>[blue,thick]{b}[ar]{(u_{1},u_{2})}
\endist
\end{istgame}
```

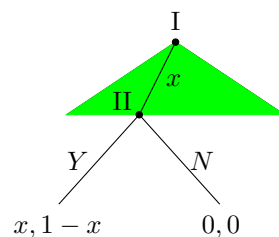
## 2.5 continuum of branches: `\istcntm`

`\istcntm`

```
% \istcntm
syntax: \istcntm[<grow key>](<coor>)(<coor>)[<fill color>]+levdist..sibdist+
default: [south]() (0,0) [black!20]+15mm..15mm+
```

Example:

```
\begin{istgame}[scale=1.2]
\istcntm(ctm)[green]+8mm..24mm+
\istroot(0)(ctm){I}+8mm..8mm+
  \istb{x}[r]
  \istb<missing>
\endist
\xtdistance{10mm}{18mm}
\istroot(1)(0-1)<[label distance=-3pt]120>{II}
  \istb{Y}[l]{x,1-x}
  \istb{N}[r]{0,0}
\endist
\end{istgame}
```



The package `istgame` provides two types of macros to represent a continuum of branches. The first type `istcntm`, in fact, draws a background with the optional color filled representing a continuum of branches (the default color is `black!25`). The second type is `\istcntm*`, which draws a background of two branches connected with an arc to represent a continuum of branches.

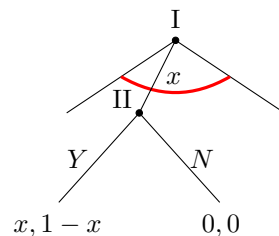
`\istcntm*`

```
% \istcntm*
syntax: \istcntm*[<grow key>](<coor>)(<coor>)[<color,opt>]{<num>}+levdist..sibdist+
default: [south]() (0,0) [black!20]{.5}+15mm..15mm+
```

The option `<num>` (0.5 in default) specifies the convex combination of the root and each background child node. The smaller `<num>` is, the closer is the connecting arc drawn to the root.

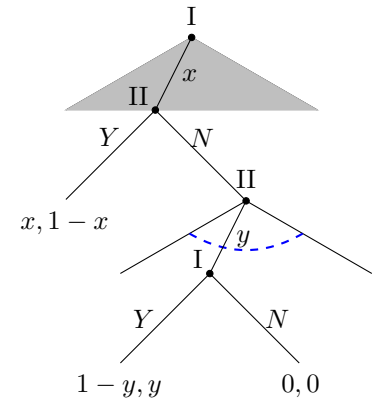
Example:

```
\begin{istgame}[scale=1.2]
\istcntm*(ctm)[very thick,red]+8mm..24mm+
\istroot(0)(ctm){I}+8mm..8mm+
  \istb{x}[right]
  \istb<missing>
\endist
\xtdistance{10mm}{18mm}
\istroot(1)(0-1)<[label distance=-3pt]120>{II}
  \istb{Y}[left]{x,1-x}
  \istb{N}[right]{0,0}
\endist
\end{istgame}
```



Example:

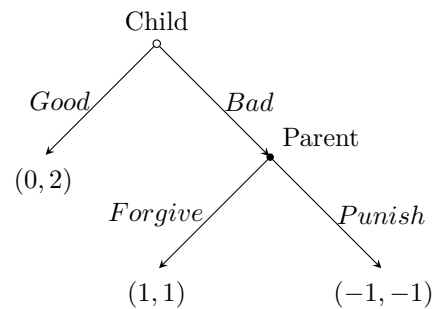
```
\begin{istgame}[scale=1.2]
\xtdistance{8mm}{2*sqrt(3)*\xtlevdist}
\istcntm(period1)
\istroot(0)(period1){I}+8mm..8mm+
\istb{x}[right]
\istb<missing>
\endist
\xtdistance{10mm}{20mm}
\istroot(1)(0-1)<[label distance=-3pt]120>{II}
\istb{Y}[above]{x,1-x}
\istb{N}[above]
\endist
%-----
\xtdistance{8mm}{3.464*\xtlevdist}
\istcntm*(period2)(1-2)[thick,dashed,blue]{.45}
\istroot(2)(period2){II}+8mm..8mm+
\istb{y}[right]
\istb<missing>
\endist
\xtdistance{10mm}{20mm}
\istroot(3)(2-1)<[label distance=-3pt]120>{I}
\istb{Y}[left]{1-y,y}
\istb{N}[right]{0,0}
\endist
\end{istgame}
```



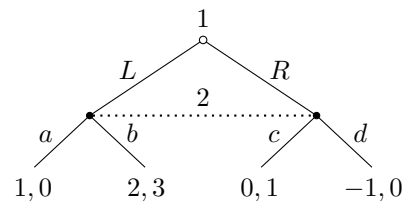
### 3 Game Tree Examples

#### 3.1 simple examples

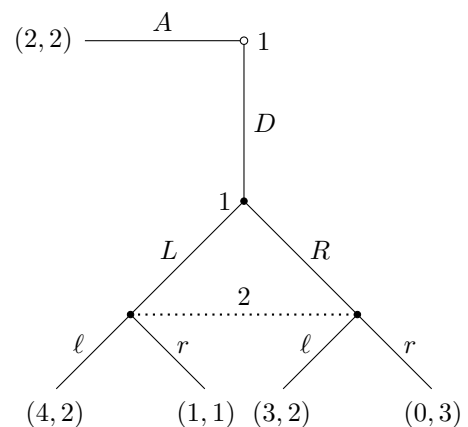
```
\begin{istgame}[->,>=stealth,shorten >=.5pt]
\xtdistance{15mm}{30mm}
\istroot(0)[initial node]{Child}
\istb{Good}[left]{(0,2)}
\istb{Bad}[right]
\endist
\istroot(1)(0-2)<30>{Parent}
\istb{Forgive}[left]{(1,1)}
\istb{Punish}[right]{(-1,-1)}
\endist
\end{istgame}
```



```
\begin{istgame}
\xtdistance{10mm}{30mm}
\istroot(0)[initial node]{1}
\istb{L}[al]
\istb{R}[ar]
\endist
\xtdistance{7mm}{15mm}
\istroot(1a)(0-1)
\istb{a}[al]{1,0}
\istb{b}[ar]{2,3}
\endist
\istroot(1b)(0-2)
\istb{c}[al]{0,1}
\istb{d}[ar]{-1,0}
\endist
\xtInfoSet(1a)(1b){2}
\end{istgame}
```

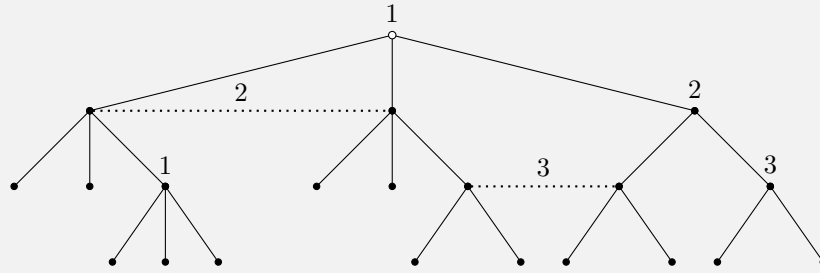


```
\begin{istgame}
\xtdistance{15mm}{30mm}
\istroot[-135](0)[initial node]<0>{1}
\istb{A}[a]{(2,2)}[l]
\istb{D}[r]
\endist
\istroot(1)(0-2)<left>{1}
\istb{L}[al]
\istb{R}[ar]
\endist
\xtInfoSet(1-1)(1-2){2}
\xtdistance{10mm}{20mm}
\istroot(2)(1-1)
\istb{\ell}[al]{(4,2)}
\istb{r}[ar]{(1,1)}
\endist
\istroot(3)(1-2)
\istb{\ell}[al]{(3,2)}
\istb{r}[ar]{(0,3)}
\endist
\end{istgame}
```



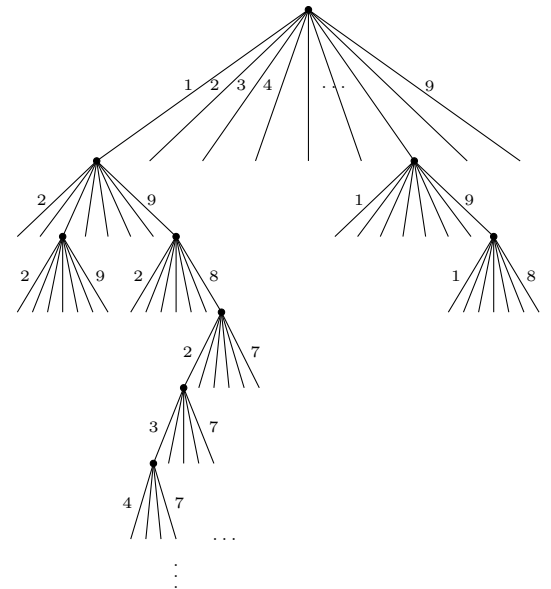


### 3.2 larger game trees



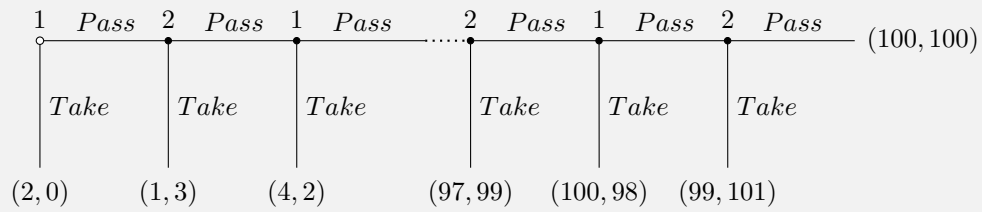
```
\begin{istgame}
\xtShowTerminalNodes
\xtdistance{10mm}{40mm}
\istroot(0)[chance node]{1} \istb \istb \istb \endist
\xtdistance{10mm}{10mm}
\istroot(1)(0-1) \istb \istb \istb \endist
\istroot(2)(0-2) \istb \istb \istb \endist
\xtdistance{10mm}{20mm}
\istroot(3)(0-3){2} \istb \istb \endist
\xtdistance{10mm}{7mm}
\istroot(a)(1-3){1} \istb \istb \istb \endist
\xtdistance{10mm}{14mm}
\istroot(b)(2-3) \istb \istb \endist
\istroot(c)(3-1) \istb \istb \endist
\istroot(d)(3-2){3} \istb \istb \endist
\xtInfoset(1)(2){2} \xtInfoset(2-3)(3-1){3}
\end{istgame}
```

```
\begin{istgame}[font=\tiny]
\xtdistance{20mm}{7mm}
\istroot(0) \istb \istb \istb \istb \istb
\istb{\dots} \istb \istb \istb{9}[r] \endist
\foreach \x in {1,...,4}
{\xtActionLabel(0)(0-\x){\x}[1]}
\xtdistance{10mm}{3mm}
\istroot(a1)(0-1) \istb{2}[1] \istb \istb \istb
\istb \istb \istb \istb{9}[r] \endist
\istroot(a7)(0-7) \istb{1}[1] \istb \istb \istb
\istb \istb \istb \istb{9}[r] \endist
\xtdistance{10mm}{2mm}
\istroot(A)(a1-3) \istb{2}[1] \istb \istb \istb
\istb \istb \istb{9}[r] \endist
\istroot(B)(a1-8) \istb{2}[1] \istb \istb \istb
\istb \istb \istb{8}[r] \endist
\istroot(C)(a7-8) \istb{1}[1] \istb \istb \istb
\istb \istb \istb{8}[r] \endist
\istroot(Bx)(B-7) \istb{2}[1] \istb \istb \istb
\istb \istb{7}[r] \endist
\istroot(By)(Bx-1) \istb{3}[1] \istb \istb \istb
\istb{7}[r] \endist
\istroot(Bz)(By-1) \istb{4}[1] \istb \istb
\istb{7}[r] \endist
\xtPayoff(Bz-4){\vdots}
\xtPayoff(Bz-4){\cdots}[xshift=10pt,right]
\end{istgame}
```



### 3.3 some more extensive games

#### 3.3.1 centipede game

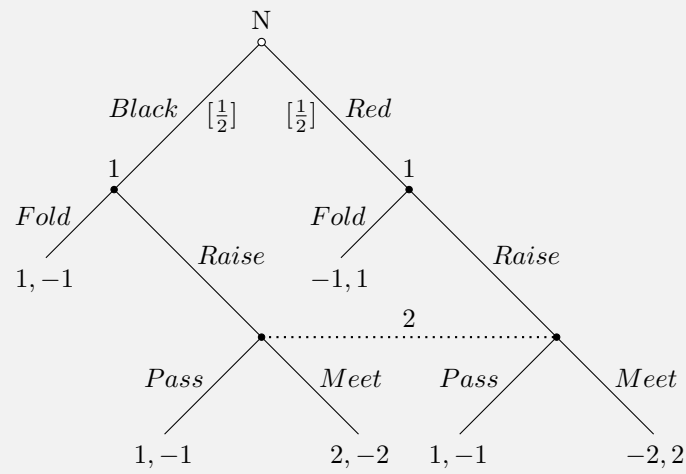


```

\begin{istgame}[scale=1.2]
\setistgrowdirection{south east}
\xtdistance{10mm}{20mm}
\istroot(0)[initial node]{1}
  \istb{Take}[r]{(2,0)}[b]
  \istb{Pass}[a]
\endist
\istroot(1)(0-2){2}
  \istb{Take}[r]{(1,3)}[b]
  \istb{Pass}[a]
\endist
\istroot(2)(1-2){1}
  \istb{Take}[r]{(4,2)}[b]
  \istb{Pass}[a]
\endist
\xtInfo{set(2-2)}{([xshift=5mm]2-2)}
%-----
\istroot(3)([xshift=5mm]2-2){2}
  \istb{Take}[r]{(97,99)}[b]
  \istb{Pass}[a]
\endist
\istroot(4)(3-2){1}
  \istb{Take}[r]{(100,98)}[b]
  \istb{Pass}[a]
\endist
\istroot(5)(4-2){2}
  \istb{Take}[r]{(99,101)}[b]
  \istb{Pass}[a]{(100,100)}[r]
\endist
\end{istgame}

```

### 3.3.2 poker game

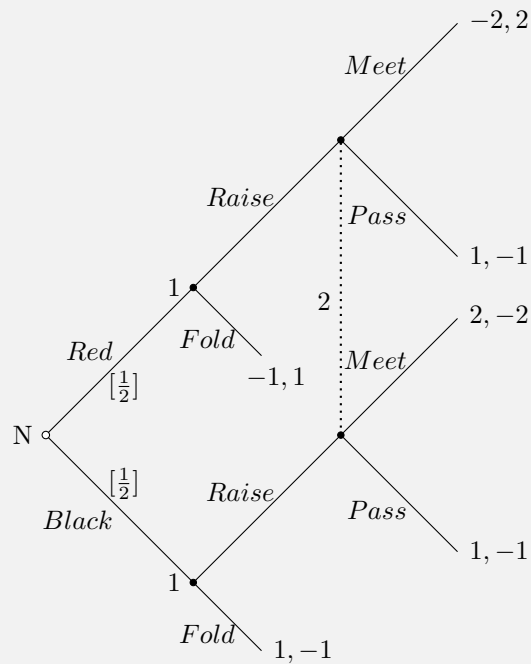


```

\begin{istgame}[scale=1.3]
\xtdistance{15mm}{30mm}
\istroot(0)[chance node]{N}
  \istb{Black}[al]
  \istb{Red}[ar]
\endist
\xtdistance{15mm}{30mm}
\istroot(1-1)(0-1){1}
  \istb<grow=-135,level distance=10mm>{Fold}[al]{1,-1}
  \istb{Raise}[ar]
\endist
\xtdistance{10mm}{20mm}
\istroot(1)(1-1-2)
  \istb{Pass}[al]{1,-1}
  \istb{Meet}[ar]{2,-2}
\endist
\xtdistance{15mm}{30mm}
\istroot(1-2)(0-2){1}
  \istb<grow=-135,level distance=10mm>{Fold}[al]{-1,1}
  \istb{Raise}[ar]
\endist
\xtdistance{10mm}{20mm}
\istroot(2)(1-2-2){}
  \istb{Pass}[al]{1,-1}
  \istb{Meet}[ar]{-2,2}
\endist
\xtInfoSet(1-1-2)(1-2-2){2}
\xtActionLabel(0)(0-1){\[\frac{1}{2}][r]
\xtActionLabel(0)(0-2){\[\frac{1}{2}][l]
\end{istgame}

```

### 3.3.3 poker game: growing to the right

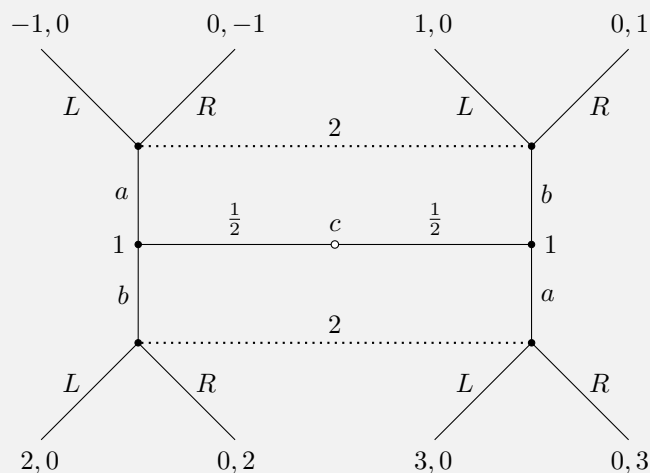


```

\begin{istgame}[scale=1.3]
\setistgrowdirection{0} % default grow-direction is 'east' from now on
\xtdistance{15mm}{30mm}
\istroot(0)[chance node]<left>{N}
  \istb{Black}[bl]
  \istb{Red}[al]
\endist
\setistactionlabelposition{6pt}{1pt} % look here
\xtdistance{15mm}{30mm}
\istroot(1-1)(0-1)<left>{1}
  \istb<grow=-45,level distance=10mm>{Fold}[bl]{1,-1}
  \istb{Raise}[al]
\endist
\xtdistance{12mm}{24mm}
\istroot(1)(1-1-2)
  \istb{Pass}[bl]{1,-1}
  \istb{Meet}[al]{2,-2}
\endist
\xtdistance{15mm}{30mm}
\istroot(1-2)(0-2)<left>{1}
  \istb<grow=-45,level distance=10mm>{Fold}[bl]{-1,1}[[xshift=5pt]below]
  \istb{Raise}[al]
\endist
\xtdistance{12mm}{24mm}
\istroot(2)(1-2-2)
  \istb{Pass}[bl]{1,-1}
  \istb{Meet}[al]{-2,2}
\endist
\xtInfoset(1-1-2)(1-2-2){2}[[left,yshift=-5pt]
\xtActionLabel(0)(0-1){\ [\frac{1}{2}]}[a]
\xtActionLabel(0)(0-2){\ [\frac{1}{2}]}[b]
\end{istgame}

```

### 3.3.4 signaling game



```

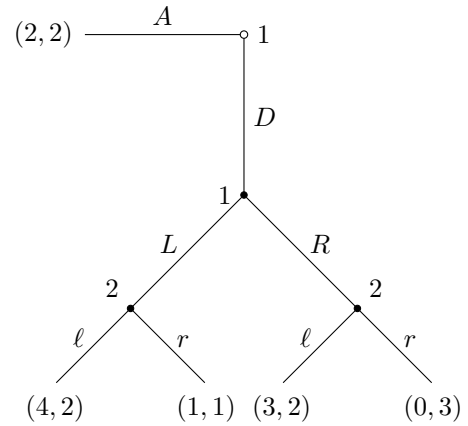
\begin{istgame}[scale=1.3]
\xtdistance{20mm}{20mm}
\istroot(0)[chance node]{$c$}
  \istb<grow=left>{\frac{1}{2}}[a]
  \istb<grow=right>{\frac{1}{2}}[a]
\endist
\xtdistance{10mm}{20mm}
\istroot(1)(0-1)<180>{1}
  \istb<grow=north>{a}[1]
  \istb<grow=south>{b}[1]
\endist
\istroot(2)(0-2)<0>{1}
  \istb<grow=south>{a}[r]
  \istb<grow=north>{b}[r]
\endist
\istroot[north](a1)(1-1)
  \istb{R}[br]{0,-1}
  \istb{L}[bl]{-1,0}
\endist
\istroot(b1)(1-2)
  \istb{L}[al]{2,0}
  \istb{R}[ar]{0,2}
\endist
\istroot(a2)(2-1)
  \istb{L}[al]{3,0}
  \istb{R}[ar]{0,3}
\endist
\istroot[north](b2)(2-2)
  \istb{R}[br]{0,1}
  \istb{L}[bl]{1,0}
\endist
\xtInfoSet(a1)(b2){2}
\xtInfoSet(b1)(a2){2}
\end{istgame}

```

### 3.4 subgames and backward induction

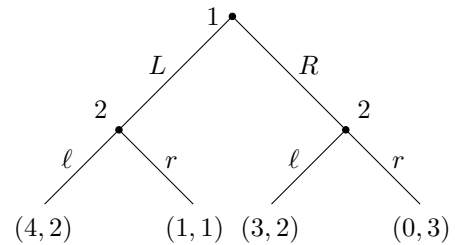
a whole game

```
\begin{istgame}
\xtdistance{15mm}{30mm}
\istroot[-135](0)[initial node]<0>{1}
\istb{A}[a]{(2,2)}[l]
\istb{D}[r]
\endist
\istroot(1)(0-2)<left>{1}
\istb{L}[al]
\istb{R}[ar]
\endist
\xtdistance{10mm}{20mm}
\istroot(2)(1-1)<135>{2}
\istb{\ell}[al]{(4,2)}
\istb{r}[ar]{(1,1)}
\endist
\istroot(3)(1-2)<45>{2}
\istb{\ell}[al]{(3,2)}
\istb{r}[ar]{(0,3)}
\endist
\end{istgame}
```



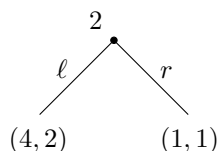
a subgame

```
\begin{istgame}
\xtdistance{15mm}{30mm}
%\istroot[-135](0)[initial node]<0>{1}
%\ \istb{A}[a]{(2,2)}[l]
%\ \istb{D}[r]
%\endist
\istroot(1)(0-2)<left>{1}
\istb{L}[al]
\istb{R}[ar]
\endist
\xtdistance{10mm}{20mm}
\istroot(2)(1-1)<135>{2}
\istb{\ell}[al]{(4,2)}
\istb{r}[ar]{(1,1)}
\endist
\istroot(3)(1-2)<45>{2}
\istb{\ell}[al]{(3,2)}
\istb{r}[ar]{(0,3)}
\endist
\end{istgame}
```



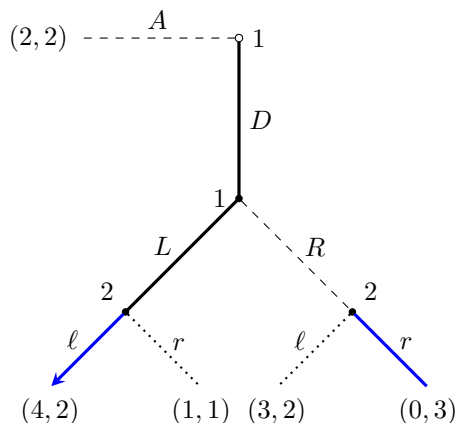
a smaller subgame

```
\begin{istgame}
%\xtdistance{15mm}{30mm}
%\istroot[-135](0)[initial node]<0>{1}
% \istb{A}[a]{(2,2)}[l]
% \istb{D}[r]
%\endist
%\istroot(1)(0-2)<left>{1}
% \istb{L}[al]
% \istb{R}[ar]
%\endist
\xtdistance{10mm}{20mm}
\istroot(2)(1-1)<135>{2}
 \istb{\ell}[al]{(4,2)}
 \istb{r}[ar]{(1,1)}
\endist
\istroot(3)(1-2)<45>{2}
% \istb{\ell}[al]{(3,2)}
% \istb{r}[ar]{(0,3)}
%\endist
\end{istgame}
```



backward induction

```
\begin{istgame}[>=stealth]
\xtdistance{15mm}{30mm}
\istroot[-135](0)[initial node]<0>{1}
 \istb[dashed]{A}[a]{(2,2)}[l]
 \istb[very thick]{D}[r]
\endist
\istroot(1)(0-2)<left>{1}
 \istb[very thick]{L}[al]
 \istb[dashed]{R}[ar]
\endist
\xtdistance{10mm}{20mm}
\istroot(2)(1-1)<135>{2}
 \istb[->,very thick,blue]{\ell}[al]{(4,2)}
 \istb[thick,dotted]{r}[ar]{(1,1)}
\endist
\istroot(3)(1-2)<45>{2}
 \istb[thick,dotted]{\ell}[al]{(3,2)}
 \istb[very thick,blue]{r}[ar]{(0,3)}
\endist
\end{istgame}
```



## Version history

- v0.91 (2017/02/13)
  - some internal macro names changed (`\xtnode`, `\xttnode`, `\xtshowtnode`)
  - conflict with `tikz-qtree` resolved. (`istgame` differs from `tikzpicture` with `tikz-qtree`)
  - `\setistnodeanchors` added
  - `zero node` added
- v0.8 (2017/01/17)
  - macro names changed (to get ready to upload to CTAN)

- \* `\xtdistance`: prefix ‘x’ changed to ‘xt’ meaning ‘extensive tree’
- \* `\xtInfoset(*)`, `\xtInfosetOwner`, `\xtActionLabel`, `xtOwner`, `\xtPayoff`, `\xtNode`
- \* `\xtShowTerminalNodes`, `\xtHideTerminalNodes`
- \* `\xtlevdist`, `\xtsibdist`
- `\xtNode` redefined
- some internal macro names changed (including `\xtpayff`, `\xtmove`)
- node styles redefined
- `\setist<Solid>NodeStyle` added for `<Various>` node styles
- `\setistdefaultnodedecolor`, `setistdefaultnodedbgcolor` added

## Acknowledgement

A special thanks goes to Kansoo Kim for his great help in using `expl3` to simplify the appearance and the usage of the macros.

## References

- Chen, H. K. (2013), “Drawing Game Trees with *TikZ*,” [http://www.sfu.ca/~haiyunc/notes/Game\\_Trees\\_with\\_TikZ.pdf](http://www.sfu.ca/~haiyunc/notes/Game_Trees_with_TikZ.pdf).
- Osborne, M. J. (2004), “Manual for `egameps.sty`,” Version 1.1, <http://texdoc.net/texmf-dist/doc/latex/egameps/egameps.pdf>
- Tantau, T. (2015), “The *TikZ* and PGF Packages,” Version 3.0.1, <http://sourceforge.net/projects/pgf>.

## Discussions (in Korean) on KTUG board

- Drawing Game Trees 1: [http://www.ktug.org/xe/index.php?document\\_srl=207287](http://www.ktug.org/xe/index.php?document_srl=207287)
- Drawing Game Trees 1-1: [http://www.ktug.org/xe/index.php?document\\_srl=207513](http://www.ktug.org/xe/index.php?document_srl=207513)
- Drawing Game Trees 1-2: [http://www.ktug.org/xe/index.php?document\\_srl=208286](http://www.ktug.org/xe/index.php?document_srl=208286)
- Drawing Game Trees 2: [http://www.ktug.org/xe/index.php?document\\_srl=212043](http://www.ktug.org/xe/index.php?document_srl=212043)
- Drawing Game Trees 2-1: [http://www.ktug.org/xe/index.php?document\\_srl=212225](http://www.ktug.org/xe/index.php?document_srl=212225)
- Drawing Game Trees 2-2: [http://www.ktug.org/xe/index.php?document\\_srl=212319](http://www.ktug.org/xe/index.php?document_srl=212319)