SemiNFG class

# SemiNFG class description:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **name** | *String* |  |
| **description** | *String* |  |
| **nodes** | *Set* | contains all *DecisionNodes* and *ChanceNodes* in the net. |
| **decision\_nodes** | *Set* | contains *DecisionNode* objects. Initialized by set\_dec\_nodes |
| **chance\_nodes** | *Set* | contains *ChanceNode* objects. Initialized by set\_chc\_nodes. |
| **edges** | *Dictionary* | keys are *String* node names, values are parent *Sets* (the parent Set is a node attribute). Initialized by set\_edges. |
| **partition** | *Dictionary* | keys are *String* player/nature names, values are *Sets* of nodes belonging to that player/nature. Initialized by set\_partition |
| **u\_functions** | *Dictionary* | keys are *String* player names, values are *Functions* from \cup X\_v \mapsto \mathbb{R} |
| **iterator** | *List* | ordered list of nodes from the *Set* self.nodes*,* used for iterating through the topological order of the net |

# SemiNFG class attributes:

# SemiNFG class methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| **\_\_init\_\_** |  |
| **add\_edge** | either a Tuple or a collection of tuples (e.g. a List of tuples), first element is parent, second element is child. Checks child’s parents attribute to make sure link exists. Calls topological\_sort, set\_partition, set\_dec\_nodes, set\_chc\_nodes, and set\_edges to update the SemiNFG |
| **add\_node** | either a DecisionNode, ChanceNode or a collection of them. Calls topological\_sort, set\_partition, set\_dec\_nodes, set\_chc\_nodes, and set\_edges to update the SemiNFG |
| **delete\_edge** | either a Tuple or a collection of tuples (e.g. a List of tuples), first element is parent, second element is child. Checks child’s parents attribute to make sure link exists. Calls topological\_sort, set\_partition, set\_dec\_nodes, set\_chc\_nodes, and set\_edges to update the SemiNFG |
| **delete\_node** | either a DecisionNode, ChanceNode or a collection of them. Calls topological\_sort, set\_partition, set\_dec\_nodes, set\_chc\_nodes, and set\_edges to update the SemiNFG |
| **ancestors** | takes a single node as input and returns a List of that node’s ancestors sorted in topological order. |
| **set\_chc\_nodes** | sets Self.chance\_nodes |
| **children** | takes a single node as input and returns a Set of that node’s children |
| **set\_dec\_nodes** | sets Self.decision\_nodes |
| **descendents** | takes a single node as input and returns a List of that node’s descendents sorted in topological order |
| **get\_leaves** | returns a Set of leaf nodes for the SemiNFG |
| **logprob** | accepts a List of values, one for each node, sorted in the topological order given by Self.iterator, and returns the corresponding joint log probability. without input, it returns the joint log probability that corresponds to the current values |
| **prob** | accepts a List of values, one for each node, sorted in the topological order given by Self.iterator, and returns the corresponding joint log probability. without input, it returns the joint log probability that corresponds to the current values |
| **get\_roots** | returns a Set of root nodes for the SemiNFG |
| **get\_values** | returns a List of current values in the topological order given by Self.iterator |
| **set\_values** | accepts a Dictionary in which keys are node names and values are node values and sets the node.value attribute to value for each node in keys. |
| **utility** | accepts a List of values, one for each node, in the topological order given by Self.iterator and returns the utility for each player. Without input, it evaluates the utility for the current values of the nodes in the net. |
| **set\_partition** | returns the Dictionary Self.partition based on the types of the nodes in the net and the player attributes of the decision nodes. |
| **topological\_sort** | returns the List Self.iterator based on the algorithm described in http://en.wikipedia.org/wiki/Topological\_sorting |
| **set\_edges** | returns the Dictionary Self.edges based on the Set of nodes in Self.nodes |
| **draw\_net** | uses networkx to draw the net given the Set self.nodes and the Dictionary self.edges |

DecisionNode Class

# DecisionNode class description:

# DecisionNode class attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **name** | *String* | the name of the decision node |
| **description** | *String* | a verbal description of the decision node. |
| **CPT** | *np.array* | gives the probability distribution over self.space given the values of self.parents |
| **parents** | *Set* | the parents of self |
| **value** | *TBD* | the current value at self. The type depends on the types of the elements of space. |
| **space** | *List* | List of the possible values at space. This is ordered because the CPT is ordered. |
| **player** | *String* | the name of the player that “owns” self. |
| **CPTdict** | *Dictionary* | keys are parent names and values are the dimension of the CPT that corresponds to that parent. |

# DecisionNode class methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| **\_\_init\_\_** |  |
| **random\_draw** | accepts Dictionary of user-supplied parent names and values and returns a value from self.space drawn randomly according to the supplied parent values and self.CPT. Without input it takes a random draw according to the current values of the parents and self.CPT |
| **randomCPT** | takes a uniform random draw of the unit simplex to generate a CPT. mixed = True (default) indicates a random mixed strategy, and mixed = False indicates a random pure strategy. |
| **perturbCPT** | generates a new CPT by perturbing the current CPT with white noise and renormalizing. The first input argument is a scale parameter of the white noise. Optional argument is a Dictionary of parents and values to determine which “sliver” of the CPT to perturb. Note that perturbing the entire CPT with noise=1 is equivalent to randomCPT and perturbing a sliver with noise=1 is equivalent to applying randomCPT to only that sliver |
| **prob** | accepts Dictionary of user-supplied parent names and values and an element of self.space and returns the corresponding element of self.CPT. Without input it returns the probability of the current values of the parents and self.value |
| **logprob** | the same inputs as prob. calls prob and evaluates np.log of the output. |
| **set\_value** | accepts an element of self.space and replaces the self.value with that input. Returns an error if the input is not an element of self.space. |
| **\_setCPTdict** | sets the Dictionary self.CPTdict by ordering the parents alphabetically by name. |
| **\_createCPT** | creates an empty np.array called self.CPT using self.CPTdict, the dimensions of the nodes in self.parents, and the size of self.space. |
| **\_check\_parents** | checks if parents are all DecisionNode or discrete ChanceNode objects so that CPT makes sense. |

ChanceNode Class

# ChanceNode class description:

# ChanceNode class attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **name** | *String* | the name of the chance node |
| **description** | *String* | a verbal description of the chance node. |
| **CPT** | *np.array* | the conditional probability distribution over self.space given the values of self.parents. This is empty if user supplies a sp.stats distribution. |
| **distribution** | *sp.stats.distribution* | Continous or discrete random variable from sp.stats. This is empty if user supplies CPT. |
| **parents** | *Set* | the parents of self. The elements are either ChanceNode or DecisionNode objects. |
| **value** | *TBD* | The current value at self. The type depends on the types of the elements of space. |
| **space** | *List* | List of the possible values at space. This is ordered because the CPT is ordered. Empty if self.continuous is True. |
| **player** | *String* | “nature” is the default value for chance nodes |
| **continuous** | *Boolean* | True if this is a continuous space. False if discrete. |
| **CPTdict** | *OrderedDict* | Supplied by the user. The key is the parent name, and the value is the parent. The order corresponds to the dimensions of the user-supplied CPT. |

# ChanceNode class methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| **\_\_init\_\_** |  |
| **random\_draw** | accepts Dictionary of user-supplied parent names and values and returns a value from self.space drawn randomly according to the supplied parent values and self.CPT. Without input it takes a random draw according to the current values of the parents and self.CPT. When user supplies an sp.stats distribution, the same is achieved via the distributions methods. |
| **prob** | accepts Dictionary of user-supplied parent names and values and an element of self.space and returns the corresponding element of self.CPT. Without input it returns the probability of the current values of the parents and self.value. When user supplies an sp.stats distribution, the same is achieved via the distributions methods. |
| **logprob** | the same inputs as prob. calls prob and evaluates np.log of the output. |
| **set\_value** | accepts an element of self.space and replaces the self.value with that input. Returns an error if the input is not an element of self.space. |
| **\_setCPTdict** | sets the Dictionary self.CPTdict by fixing the parents in a dictionary in the order set by the parents list. |

seminfg functions

|  |  |
| --- | --- |
| **function** | **Description** |
| check\_parents | makes sure the parents are either DecisionNodes or discrete ChanceNodes |