API Documentation

API Documentation

July 27, 2014

Contents

1 Module splacris

1.1 Variables

Name	Description
__package__	**Value:** `None`

1.2 Class Cost

This class loads the cost table to a Python hash table the hash is <feature><trace list> i.e. "F["K","L","U"]":5.5

1.2.1 Methods

__init__(*self*, *cost_file*)

Constructor method for Cost class , This method instance a cost object

Parameters

`self:` the cost table instance
(type=Cost instance)

`cost_file:` Cost table file
(type=string)

Return Value

A Cost object
(type=Cost object)

get_cost(*self*, *trace*, *feature*)

Function to get the cost of compute a feature given the previously computed features

Parameters

`self:` the current Cost instance
(type=Cost instance)

`trace:` The features trace
(type=list)

`feature:` The produced feature
(type=string)

Return Value

The cost of compute the feature
(type=float)

1.3 Class SPLAcris

This class defines the SPLAcris cost function In this case a SPLAcris object will represent a state in the FSM

1.3.1 Methods

__init__(*self*, *term*, *trace*, *cost*, *itime*, *etime*, *uthreshold*, *lthreshold*)

Constructor method for SPLAcris class , This method instance a SPLAcris object

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`term:` Current SPLAcris term
(type=etree SPLAcris term)

`trace:` Computed trace to generate current term
(type=string)

`cost:` Cost to generate current trace
(type=integer)

`itime:` Initial time
(type=time)

`etime:` End time
(type=time)

`uthreshold:` Upper threshold
(type=float)

`lthreshold:` Lower threshold
(type=float)

Return Value

A SPLAcris object
(type=SPLAcris object)

check_syntax(*self*)

Function to check the term syntax

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

Return Value

1 if is good 0 if bad
(type=integer)

check_constraints(*self*, *tree_original*, *feature*)

Function to check if a constraint can be computed this method returns a tree with the contraint computed

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`tree_original:` The actual SPLA term
(type=etree)

`feature:` The produced feature (to check the constraint)
(type=string)

Return Value

The new tree with the constraint computed if applies
(type=etree SPLAcris term)

is_forb(*self*, *tree*, *feature*)

Function to check if a feature is forbidden this method returns 1 if the feature is forbidden, 0 otherwise

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`tree:` The actual SPLA term
(type=etree)

`feature:` The produced feature
(type=string)

Return Value

1 if is forbidden 0 otherwise
(type=integer)

is_mand(*self*, *tree*, *feature*)

Function to check if a feature is mandatory this method returns 1 if the feature is mandatory, 0 otherwise

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

`feature:` The produced feature

(type=string)

Return Value

1 if is forbidden 0 otherwise

(type=integer)

is_optional(*self*, *tree*, *feature*)

Function to check if a feature is optional

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

`feature:` searched feature

(type=string)

Return Value

1 if optional 0 otherwise

(type=integer)

is_feature_inside(*self*, *tree*, *feature*)

Function to check if a feature is inside the term

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

`feature:` searched feature

(type=string)

Return Value

1 if is found 0 otherwise

(type=integer)

has_nil(*self*, *tree*)

Function to check if the term has nil inside

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=tree)

Return Value

1 if a nil is found, 0 otherwise

(type=integer)

is_computable(*self*, *tree*, *feature*)

Function to check if a feature is computable

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

`feature:` searched feature

(type=string)

Return Value

0 if is not computable 1 otherwise

(type=integer)

is_computed(*self*, *trace*, *feature*)

Function to check if a feature was computed

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`trace:` Computed features

(type=list)

`feature:` searched feature

(type=string)

Return Value

0 if was not computed 1 if yes

(type=integer)

has_mand(*self*, *tree*)

Function to check if a term has mand features this method returns 1 if has mand features, 0 otherwise

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

Return Value

1 if is has mand features 0 otherwise

(type=integer)

get_mand(*self*, *tree*)

Function to get the mand features list this method returns a list of mand features

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

Return Value

a list with mand features

(type=list)

get_forb(*self*, *tree*)

Function to get the forb features list this method returns a list of forb features

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

Return Value

a list with forb features

(type=list)

is_blocked(*self*, *tree*)

get_computables(*self*, *tree*)

Function to get the computables list

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree:` The actual SPLA term

(type=etree)

Return Value

list with computables features

(type=list)

merge_parallels(*self*)

Function to merge consecutive parallels in order to be able to process the term

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

Return Value

1 if parallels were merge, 0 otherwise

(type=boolean)

merge_choose_1(*self*)

Function to merge consecutive choose_1 in order to be able to process the term

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

Return Value

1 if choose_1 were merge, 0 otherwise

(type=boolean)

remove_choose_1(*self*, *tree_original*, *feature*)

Function to remove features inside the choose_1 operator and leave the feature to be computed with the choose_1

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`tree_original:` The source tree
(type=etree SPLAcris term)

`feature:` The feature which indicates which choose_1 must be removed
(type=string)

Return Value

An etree object without the choose_1 operator
(type=etree SPLAcris term)

remove_lonely_choose_1(*self*, *tree_original*)

Function to remove lonely choose_1 relationships

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`tree_original:` The actual SPLA term
(type=etree)

Return Value

tree without lonely choose_1 relationships
(type=etree)

remove_feature(*self*, *tree_original*, *feature*)

Function to produce a feature, receives a term and a feature to be extracted and returns the term without the feature

Parameters

`self:` the current SPLAcris instance
(type=SPLAcris instance)

`tree_original:` The source tree
(type=etree SPLAcris term)

`feature:` The feature to be removed
(type=string)

Return Value

An etree object without the feature
(type=etree SPLAcris term)

can_tick(*self*, *term*)

Function to check if a term can tick It will check for: TICK - OFEAT2 - CON3 - REQ3 - EXCL4 - FORB2 - MAND1

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`term:` The actual SPLA term

(type=etree)

Return Value

1 if can tick 0 otherwise

(type=integer)

compute_feature(*self*, *tree_original*, *feature*, *cost_table*)

Function to compute a feature, receives a SPLAcris instance, a feature to be computed and a cost table

Parameters

`self:` the current SPLAcris instance

(type=SPLAcris instance)

`tree_original:` The source tree

(type=etree SPLAcris term)

`feature:` The feature to be removed

(type=string)

`cost_table:` The cost table

(type=A cost function object)

Return Value

A SPLAcris object

(type=SPLAcris)

Index