hybrid

Release 0.0.1

Iain Moncrief,Nick Burrell

Feb 09, 2022

CONTENTS:

CHAPTER

ONE

API REFERENCE

This page contains auto-generated API reference documentation[1].

1.1 hybrid

1.1.1 Submodules

hybrid.language

Module Contents

Functions

definition_libraries(ast)	→ list[hybrid.syntax.LibraryDefinition]
definition_calling_programs(ast)	→ list[hybrid.syntax.CallingProgramDefinition]

hybrid.language.**definition_libraries**(*ast*) → list[*hybrid.syntax.LibraryDefinition*]

hybrid.language.**definition_calling_programs**(*ast*) → list[*hybrid.syntax.CallingProgramDefinition*]

hybrid.latex

Module Contents

Classes

Stylizer
Relation
HighlightExpression

continues on next page

[1] Created with sphinx-autoapi

Table 2 – continued from previous page

Link
Interchangable
Indistinguishable

Functions

bin_op_tex(op: str) → str
texify_relation(relation: str) → str
texify(ast: hybrid.syntax.AST, queries_only=True) → str
link(prog: hybrid.syntax.AST, lib: hybrid.syntax.LibraryDefinition, glowing=False) → str
equiv(left: hybrid.syntax.AST, right: hybrid.syntax.AST, align=False) → str
save_file(tex_source, file)

Attributes

context
is_in_math_block
depth

class hybrid.latex.**Stylizer**
Bases: *hybrid.syntax.AST*

class hybrid.latex.**Relation**
Bases: *Stylizer*

class hybrid.latex.**HighlightExpression**
Bases: *Stylizer*

expression :hybrid.syntax.AST

class hybrid.latex.**Link**
Bases: *Relation*

lhs :hybrid.syntax.AST

rhs :hybrid.syntax.AST

class hybrid.latex.**Interchangable**
Bases: *Relation*

lhs :hybrid.syntax.AST

rhs :hybrid.syntax.AST

class hybrid.latex.**Indistinguishable**

Bases: *Relation*

lhs :hybrid.syntax.AST

rhs :hybrid.syntax.AST

hybrid.latex.**bin_op_tex**(*op: str*) → str

hybrid.latex.**texify_relation**(*relation: str*) → str

hybrid.latex.**context**

hybrid.latex.**is_in_math_block = False**

hybrid.latex.**depth = 0**

hybrid.latex.**texify**(*ast:* hybrid.syntax.AST, *queries_only=True*) → str

hybrid.latex.**link**(*prog:* hybrid.syntax.AST, *lib:* hybrid.syntax.LibraryDefinition, *glowing=False*) → str

hybrid.latex.**equiv**(*left:* hybrid.syntax.AST, *right:* hybrid.syntax.AST, *align=False*) → str

hybrid.latex.**save_file**(*tex_source*, *file*)

hybrid.parser

Module Contents

Classes

RawSyntaxTreeVisitor

Functions

get_rule_name(node: antlr4.tree.Tree.Tree) → str
get_children_derivation_names(node: antlr4.tree.Tree.Tree)
copy_context(node)
expand_all_derivations(node: antlr4.ParserRuleContext.ParserRuleContext)
non_terminal_children(ctx)
file_to_ast_dict(file)

Attributes

NonTerminalName

hybrid.parser.**NonTerminalName**

hybrid.parser.**get_rule_name**(*node: antlr4.tree.Tree.Tree*) → str

hybrid.parser.**get_children_derivation_names**(*node: antlr4.tree.Tree.Tree*)

hybrid.parser.**copy_context**(*node*)

hybrid.parser.**expand_all_derivations**(*node: antlr4.ParserRuleContext.ParserRuleContext*)

hybrid.parser.**non_terminal_children**(*ctx*)

class hybrid.parser.**RawSyntaxTreeVisitor**

Bases: hybrid.parsing.HybLangVisitor.HybLangVisitor

visitProgram(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.ProgramContext*)

visitProcedure(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.ProcedureContext*)

visitCall_prog_def(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Call_prog_defContext*)

visitLibrary_def(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Library_defContext*)

visitBlock(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.BlockContext*)

visitStatement(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.StatementContext*)

visitAssignment(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.AssignmentContext*)

visitSample(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.SampleContext*)

visitSet_type(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Set_typeContext*)

visitSet_lit(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Set_litContext*)

visitExpression(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.ExpressionContext*)

visitBin_op(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Bin_opContext*)

visitFunction_call(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Function_callContext*)

visitReturn_stmt(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Return_stmtContext*)

visitFunction_arguments(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Function_argumentsContext*)

visitFunction_argument(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Function_argumentContext*)

visitFunction_def(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Function_defContext*)

visitTerminal(*self*, *node: antlr4.tree.Tree.TerminalNodeImpl*)

visitIf_stmt(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.If_stmtContext*)

visitQuery_statement(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Query_statementContext*)

visitShow_query_statement(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Show_query_statementContext*)

visitQuery_relation(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Query_relationContext*)

visitQuery_expression(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Query_expressionContext*)

visitString_literal(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.String_literalContext*)

visitWrite_query_statement(*self*, *ctx: hybrid.parsing.HybLangParser.HybLangParser.Write_query_statementContext*)

aggregateResult(*self*, *aggregate*, *nextResult*)

defaultResult(*self*)

hybrid.parser.**file_to_ast_dict**(*file*)

hybrid.syntax

Module Contents

Classes

AST
Statement
NonStatement
Block
Definitions
Expression
Literal
Identifier
Number
String
BinaryOperation
FunctionCall
Assignment
Sample
Return
SetAtom

continues on next page

Table 8 – continued from previous page

NumberSetAtom
IdentifierSetAtom
SetLiteral
FunctionArgument
FunctionArguments
FunctionDefinition
IfStatement
CallingProgramDefinition
LibraryDefinition
QueryExpression
QueryIdentifier
QueryStatement
ProgramReference
LibraryReference
QueryRelation
Write
ShowQueryStatement
WriteQueryStatement

Functions

construct_ast(parse_tree) → Any

class hybrid.syntax.**AST**

class hybrid.syntax.**Statement**
Bases: *AST*

class hybrid.syntax.**NonStatement**
Bases: *AST*

class hybrid.syntax.**Block**
Bases: *AST*

procedure :list[Statement | NonStatement]

class hybrid.syntax.**Definitions**
Bases: *AST*

body :Block

class hybrid.syntax.**Expression**
Bases: *Statement*

class hybrid.syntax.**Literal**
Bases: *Expression*

class hybrid.syntax.**Identifier**
Bases: *Literal*

id :str

class hybrid.syntax.**Number**
Bases: *Literal*

num :int

class hybrid.syntax.**String**
Bases: *Literal*

string :str

class hybrid.syntax.**BinaryOperation**
Bases: *Expression*

lhs :Expression

bin_op :str

rhs :Expression

class hybrid.syntax.**FunctionCall**
Bases: *Expression*

fun_name :str

args :list[Expression]

class hybrid.syntax.**Assignment**
Bases: *Statement*

lhs :Identifier

rhs :Expression

class hybrid.syntax.**Sample**
Bases: *Statement*

lhs :Identifier

rhs :Expression | Any

class hybrid.syntax.**Return**
Bases: *Statement*

ret :Expression

class hybrid.syntax.**SetAtom**
Bases: *AST*

class hybrid.syntax.**NumberSetAtom**
Bases: *SetAtom*

num :int

class hybrid.syntax.**IdentifierSetAtom**
Bases: *SetAtom*

id :str

class hybrid.syntax.**SetLiteral**
Bases: *AST*

data :list[SetAtom]

class hybrid.syntax.**FunctionArgument**
Bases: *AST*

arg_id :str

arg_type :Optional[SetLiteral]

class hybrid.syntax.**FunctionArguments**
Bases: *AST*

args :list[tuple[int, FunctionArgument]]

class hybrid.syntax.**FunctionDefinition**
Bases: *NonStatement*

fun_name :str

args :FunctionArguments

block :Block

class hybrid.syntax.**IfStatement**
Bases: *NonStatement*

condition :Expression

then_block :Block

else_block :Optional[Block]

class hybrid.syntax.**CallingProgramDefinition**
Bases: *NonStatement*

name :str

body :Block

class hybrid.syntax.**LibraryDefinition**
Bases: *NonStatement*

name :str

exposing :list[tuple[str, str]]

body :Block

class hybrid.syntax.**QueryExpression**
Bases: *AST*

class hybrid.syntax.**QueryIdentifier**
Bases: *QueryExpression*

class hybrid.syntax.**QueryStatement**
Bases: *NonStatement*

class hybrid.syntax.**ProgramReference**
Bases: *QueryIdentifier*

name :str

class hybrid.syntax.**LibraryReference**
Bases: *QueryIdentifier*

name :str

class hybrid.syntax.**QueryRelation**
Bases: *QueryExpression*

lhs :QueryExpression

relation :str

rhs :QueryExpression

class hybrid.syntax.**Write**
Bases: *QueryExpression*

body :String

class hybrid.syntax.**ShowQueryStatement**
Bases: *QueryStatement*

expression :QueryExpression

class hybrid.syntax.**WriteQueryStatement**
Bases: *QueryStatement*

body :Write

hybrid.syntax.**construct_ast**(*parse_tree*) → Any

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

h

INDEX

L

M

N

P

Q

R

S

T

V

W