

---

# **Rewrite**

***Release 0.1***

**SymCollab**

**Nov 27, 2020**



## CONTENTS:

<b>1 Rewrite Library</b>	<b>1</b>
<b>2 Variants Library</b>	<b>3</b>
<b>Python Module Index</b>	<b>5</b>
<b>Index</b>	<b>7</b>



**REWRITE LIBRARY**



## VARIANTS LIBRARY

The variants module is responsible for computing variants and identifying some properties about them.

**class** `rewrite.variants.Variants` (*term: Union[algebra.term.Variable, algebra.term.Constant, algebra.term.FuncTerm], rules: rewrite.system.RewriteSystem*)

Construct variants of a term given a rewrite system.

**term** [Term] The term to compute the variants from.

**rules** [RewriteSystem] The rules from which to compute the variants from.

A variant of a term `$t$` is a term that can be obtained by applying a sequence of rewrite rules to `$t$`.

```

>>> from algebra import Constant, Function, Variable
>>> from rewrite import RewriteRule, RewriteSystem, Variants
>>> f = Function("f", 2)
>>> x = Variable("x")
>>> a = Constant("a")
>>> b = Constant("b")
>>> r1 = RewriteRule(f(x, x), x)
>>> r2 = RewriteRule(f(a, x), b)
>>> term = f(a, f(b, b))
>>> vt = Variants(term, RewriteSystem({r1, r2}))
>>> list(vt)
[f(a, f(b, b)), b, f(a, b)]

```

`rewrite.variants.is_finite` (*v: rewrite.variants.Variants, bound: int = -1*)  $\rightarrow$  bool

Check to see if there are a finite number of variants.

Returns false if the variants are infinite or the bound is reached.

**v** [Variants] The variants in which to check if it's finite.

**bound** [int] The bound at which to stop checking for variants. More specifically, the bound represents the maximum number of rewrite rules that the program is checking for. An infinite bound can be specified with -1.

`rewrite.variants.narrow` (*term: Union[algebra.term.Variable, algebra.term.Constant, algebra.term.FuncTerm], goal\_term: Union[algebra.term.Variable, algebra.term.Constant, algebra.term.FuncTerm], rules: rewrite.system.RewriteSystem, bound: int = -1*)  $\rightarrow$  Optional[List[Tuple[rewrite.rule.RewriteRule, str]]]

Returns the sequence of rewrite rules necessary to rewrite one term to a goal term. If the term cannot be rewritten, this function will return None. A bound greater than -1 will set the function to stop attempting to reach the goal after a certain number of steps.

**term** [Term] The term to start from.

**goal\_term** [Term] The term to attempt to rewrite to.

**rules** [RewriteSystem] The rules from which to rewrite from.

**bound** [int] The maximum number of times to attempt rewriting. -1 indicates an infinite bound.

```
>>> from algebra import Constant, Function, Variable
>>> from rewrite import RewriteRule, RewriteSystem, narrow
>>> f = Function("f", 2)
>>> x = Variable("x")
>>> a = Constant("a")
>>> b = Constant("b")
>>> r1 = RewriteRule(f(x, x), x)
>>> r2 = RewriteRule(f(a, x), b)
>>> term = f(a, f(b, b))
>>> narrow(term, f(a,b), RewriteSystem({r1, r2}), -1)
[(f(x, x) → x, '2')]
```



## PYTHON MODULE INDEX

r

`rewrite.variants`, 3



## INDEX

### I

`is_finite()` (*in module `rewrite.variants`*), 3

### M

module

`rewrite.variants`, 3

### N

`narrow()` (*in module `rewrite.variants`*), 3

### R

`rewrite.variants`

    module, 3

### V

`Variants` (*class in `rewrite.variants`*), 3