Standard Semirings

Release 10.1

The Sage Development Team

Aug 21, 2023
## CONTENTS

1 Non Negative Integer Semiring  
2 Tropical Semirings  
3 Indices and Tables  
Python Module Index  
Index
**Class** `sage.rings.semirings.non_negative_integer_semiring.NonNegativeIntegerSemiring`  
**Bases:** `NonNegativeIntegers`  
A class for the semiring of the non negative integers  
This parent inherits from the infinite enumerated set of non negative integers and endows it with its natural semiring structure.  
**Examples:**

```python
sage: NonNegativeIntegerSemiring()
Non negative integer semiring
```

For convenience, `NN` is a shortcut for `NonNegativeIntegerSemiring()`:

```python
sage: NN == NonNegativeIntegerSemiring()
True
```

```python
sage: NN.category()
Category of facade infinite enumerated commutative semirings
```

Here is a piece of the Cayley graph for the multiplicative structure:

```python
sage: G = NN.cayley_graph(elements=range(9), generators=[0,1,2,3,5,7]) # optional - sage.graphs
```

```python
sage: G
Looped multi-digraph on 9 vertices
```

```python
sage: G.plot() # optional - sage.graphs
Graphics object consisting of 48 graphics primitives
```

This is the Hasse diagram of the divisibility order on `NN`.

```python
sage: Poset(NN.cayley_graph(elements=[1..12], generators=[2,3,5,7,11])).show() # optional - sage.combinat sage.graphs
```

Note: as for `NonNegativeIntegers`, `NN` is currently just a “facade” parent; namely its elements are plain Sage `Integers` with `Integer` Ring as parent:

```python
sage: x = NN(15); type(x)
<class 'sage.rings.integer.Integer'>
sage: x.parent()
```

(continues on next page)
Integer Ring

```
sage: x+3
18
```

**additive_semigroup_generators()**

Returns the additive semigroup generators of `self`.

**EXAMPLES:**

```
sage: NN.additive_semigroup_generators()
Family (0, 1)
```
TROPICAL SEMIRINGS

AUTHORS:

• Travis Scrimshaw (2013-04-28) - Initial version

class sage.rings.semirings.tropical_semiring.TropicalSemiring(base, use_min=True)

Bases: Parent, UniqueRepresentation

The tropical semiring.

Given an ordered additive semigroup \( R \), we define the tropical semiring \( T = R \cup \{+\infty\} \) by defining tropical addition and multiplication as follows:

\[
a \oplus b = \min(a, b), \quad a \oslash b = a + b.
\]

In particular, note that there are no (tropical) additive inverses (except for \( \infty \)), and every element in \( R \) has a (tropical) multiplicative inverse.

There is an alternative definition where we define \( T = R \cup \{-\infty\} \) and alter tropical addition to be defined by

\[
a \oplus b = \max(a, b).
\]

To use the \( \max \) definition, set the argument \( \text{use}_\text{min} = \text{False} \).

Warning: \( \text{zero}() \) and \( \text{one}() \) refer to the tropical additive and multiplicative identities respectively. These are not the same as calling \( T(0) \) and \( T(1) \) respectively as these are not the tropical additive and multiplicative identities respectively.

Specifically do not use \( \text{sum}(\ldots) \) as this converts 0 to 0 as a tropical element, which is not the same as \( \text{zero}() \). Instead use the \( \text{sum} \) method of the tropical semiring:

```
sage: T = TropicalSemiring(QQ)

sage: sum([T(1), T(2)]) # This is wrong
0
sage: T.sum([T(1), T(2)]) # This is correct
1
```

Be careful about using code that has not been checked for tropical safety.

INPUT:

• \text{base} – the base ordered additive semigroup \( R \)

• \text{use}_\text{min} – (default: \text{True}) if \text{True}, then the semiring uses \( a \oplus b = \min(a, b) \); otherwise uses \( a \oplus b = \max(a, b) \)
EXAMPLES:

```
sage: T = TropicalSemiring(QQ)
sage: elt = T(2); elt
2
Recall that tropical addition is the minimum of two elements:
```
```
sage: T(3) + T(5)
3
```

Tropical multiplication is the addition of two elements:
```
sage: T(2) * T(3)
5
sage: T(0) * T(-2)
-2
```

We can also do tropical division and arbitrary tropical exponentiation:
```
sage: T(2) / T(1)
1
sage: T(2)^(-3/7)
-6/7
```

Note that “zero” and “one” are the additive and multiplicative identities of the tropical semiring. In general, they are not the elements 0 and 1 of \( R \), respectively, even if such elements exist (e.g., for \( R = \mathbb{Z} \)), but instead the (tropical) additive and multiplicative identities \(+\infty\) and 0 respectively:
```
sage: T.zero() + T(3) == T(3)
True
sage: T.one() * T(3) == T(3)
True
sage: T.zero() == T(0)
False
sage: T.one() == T(1)
False
```

**Element**

alias of *TropicalSemiringElement*

**additive_identity()**

Return the (tropical) additive identity element \(+\infty\).

EXAMPLES:
```
sage: T = TropicalSemiring(QQ)
sage: T.zero()
+infinity
```

**gens()**

Return the generators of *self*.

EXAMPLES:
sage: T = TropicalSemiring(QQ)
sage: T.gens()
(1, +infinity)

infinity()
Return the (tropical) additive identity element $+\infty$.
EXAMPLES:

sage: T = TropicalSemiring(QQ)
sage: T.zero()
+infinity

multiplicative_identity()
Return the (tropical) multiplicative identity element 0.
EXAMPLES:

sage: T = TropicalSemiring(QQ)
sage: T.one()
0

one()
Return the (tropical) multiplicative identity element 0.
EXAMPLES:

sage: T = TropicalSemiring(QQ)
sage: T.one()
0

zero()
Return the (tropical) additive identity element $+\infty$.
EXAMPLES:

sage: T = TropicalSemiring(QQ)
sage: T.zero()
+infinity

class sage.rings.semirings.tropical_semiring.TropicalSemiringElement
Bases: Element
An element in the tropical semiring over an ordered additive semigroup $R$. Either in $R$ or $\infty$. The operators $+,. \cdot$ are defined as the tropical operators $\oplus, \odot$ respectively.

lift()
Return the value of self lifted to the base.
EXAMPLES:

sage: T = TropicalSemiring(QQ)
sage: elt = T(2)
sage: elt.lift()
2
sage: elt.lift().parent() is QQ
True
True
\texttt{sage}: \texttt{T.additive_identity().lift().parent()}
The Infinity Ring

\textbf{multiplicative_order()}

Return the multiplicative order of \texttt{self}.

\textbf{EXAMPLES:}

\begin{verbatim}
\texttt{sage}: T = TropicalSemiring(QQ)
\texttt{sage}: T.multiplicative_identity().multiplicative_order()
1
\texttt{sage}: T.additive_identity().multiplicative_order()
+Infinity
\end{verbatim}

class \texttt{sage.rings.semirings.tropical_semiring.TropicalToTropical}

\textbf{Bases:} \texttt{Map}

Map from the tropical semiring to itself (possibly with different bases). Used in coercion.
• Index
• Module Index
• Search Page
sage.rings.semirings.non_negative_integer_semiring, 1
sage.rings.semirings.tropical_semiring, 3
<table>
<thead>
<tr>
<th>A</th>
<th>additive_identity() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>one() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 5</td>
</tr>
<tr>
<td>E</td>
<td>Element (sage.rings.semirings.tropical_semiring.TropicalSemiring attribute), 4</td>
</tr>
<tr>
<td>S</td>
<td>NonNegativeIntegerSemiring (class in sage.rings.semirings.non_negative_integer_semiring), 1</td>
</tr>
<tr>
<td>G</td>
<td>gens() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 4</td>
</tr>
<tr>
<td>S</td>
<td>sage.rings.semirings.tropical_semiring module, 3</td>
</tr>
<tr>
<td>L</td>
<td>lift() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 5</td>
</tr>
<tr>
<td>T</td>
<td>TropicalSemiring (class in sage.rings.semirings.tropical_semiring), 3</td>
</tr>
<tr>
<td>Z</td>
<td>zero() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 5</td>
</tr>
<tr>
<td>N</td>
<td>NonNegativeIntegerSemiring (class in sage.rings.semirings.non_negative_integer_semiring), 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>module</th>
</tr>
</thead>
<tbody>
<tr>
<td>sage.rings.semirings.non_negative_integer_semiring, 1</td>
<td></td>
</tr>
<tr>
<td>sage.rings.semirings.tropical_semiring, 3</td>
<td></td>
</tr>
<tr>
<td>multiplicative_identity() (sage.rings.semirings.tropical_semiring.TropicalSemiring method), 5</td>
<td></td>
</tr>
<tr>
<td>multiplicative_order() (sage.rings.semirings.tropical_semiring.TropicalSemiringElement method), 6</td>
<td></td>
</tr>
</tbody>
</table>