Why Sage?

Features we have and want in combinatorics in Sage

Anne Schilling, UC Davis

IMA, Minnesota, August 21, 2017
Why Sage?
Features we have and want in combinatorics in Sage

Anne Schilling, UC Davis

IMA, Minnesota, August 21, 2017

Eclipse
A Story

One of my passions are crystal bases which provide a combinatorial tool to study algebraic/geometric structures such as:

- quantum groups
- affine Schubert calculus
- symmetric functions
- representation theory

Combinatorics lends itself to computational analysis!
One of my passions are

**crystal bases** which provide a **combinatorial** tool to study
algebraic/geometric structures such as
- quantum groups
- affine Schubert calculus
- symmetric functions
- representation theory

**Combinatorics** lends itself to computational analysis!
A Story

One of my passions are

- **crystal bases** which provide a **combinatorial** tool to study
  - algebraic/geometric structures such as
    - quantum groups
    - affine Schubert calculus
    - symmetric functions
    - representation theory

**Combinatorics** lends itself to computational analysis!
A Story

One of my passions are

crystal bases which provide a combinatorial tool to study algebraic/geometric structures such as

- quantum groups
- affine Schubert calculus
- symmetric functions
- representation theory

Combinatorics lends itself to computational analysis!
A Story

One of my passions are

- **crystal bases** which provide a **combinatorial** tool to study
- **algebraic/geometric structures** such as
  - quantum groups
  - affine Schubert calculus
  - symmetric functions
  - representation theory

**Combinatorics** lends itself to computational analysis!
A Story

One of my passions are

crystal bases which provide a combinatorial tool to study
algebraic/geometric structures such as
• quantum groups
• affine Schubert calculus
• symmetric functions
• representation theory

Combinatorics lends itself to computational analysis!
Crystals

\[ B(\square) \]

\[ B(\Box) \]
Lonely Programming ...

- Programmed crystals in Mathematica
- Programmed what I needed right then for research
- No tests or documentation
- Could not reuse my own code a few weeks later (forgot how it worked ...)
- Kept writing similar code over and over
Lonely Programming ...

- Programmed crystals in Mathematica
- Programmed what I needed right then for research
- No tests or documentation
- Could not reuse my own code a few weeks later (forgot how it worked ...)
- Kept writing similar code over and over
Lonely Programming ...

- Programmed crystals in Mathematica
- Programmed what I needed right then for research
- No tests or documentation
- Could not reuse my own code a few weeks later (forgot how it worked ...)
- Kept writing similar code over and over
Lonely Programming ...

- Programmed crystals in Mathematica
- Programmed what I needed right then for research
- No tests or documentation
- Could not reuse my own code a few weeks later (forgot how it worked ...)
- Kept writing similar code over and over
Lonely Programming ...

- Programmed crystals in Mathematica
- Programmed what I needed right then for research
- No tests or documentation
- Could not reuse my own code a few weeks later (forgot how it worked ...)
- Kept writing similar code over and over
Sage Days 7 at IPAM in 2008

with Nicolas Thiéry started porting crystal code to Sage

Dan Bump uses crystals in number theory

William Stein  Franco Saliola  Mike Hansen  Mike Zabrocki
What can Sage do?

```python
sage: B = crystals.Tableaux(['A', 2], shape=[2, 1])
sage: u = B.highest_weight_vector(); u
[[1, 1], [2]]
sage: b = u.f(1); b
[[1, 2], [2]]
sage: type(b)
<class 'sage.combinat.crystals.tensor_product.CrystalOfTableaux_with_category.element_class'>
sage: u.weight()
(2, 1, 0)
sage: b.weight()
(1, 2, 0)
```

But the following does not work ...

```python
sage: b.shape()
--------------------------------------------------------------------
AttributeError: 'CrystalOfTableaux_with_category.element_class' object has no attribute 'shape'
sage: b.to_tableau().shape()
[2, 1]
```
Implementation of a crystal

class HighestWeightCrystalOfTypeA(UniqueRepresentation, Parent):

def __init__(self, n = 3):
    Parent.__init__(self, category = ClassicalCrystals())
    self.n = n
    self._cartan_type = CartanType(['A',n])
    self.module_generators = [ self(1) ]

def __repr__(self):
    return "Highest weight crystal of type A_%s of highest weight omega_1"%(self.n)

class Element(ElementWrapper):

def e(self, i):
    if self.value == i+1:
        return self.parent()(self.value-1)
    else:
        return None

def f(self, i):
    if self.value == i:
        return self.parent()(self.value+1)
    else:
        return None
Moral of the Story ...
End/beginning of the Story ...

Semester long program at ICERM on Automorphic Forms, Combinatorial Representation Theory and Multiple Dirichlet Series, Spring 2013

Thematic Tutorial: Lie Methods and Related Combinatorics in Sage

$k$-Schur functions and affine Schubert calculus
Moral of the Story ...

End/beginning of the Story ...

Semester long program at ICERM on
Automorphic Forms, Combinatorial Representation Theory
and Multiple Dirichlet Series, Spring 2013

Thematic Tutorial: Lie Methods and Related Combinatorics in Sage

$k$-Schur functions and affine Schubert calculus
Wishlist

- trac #23560 Benkart-Kang-Kashiwara crystals for super Lie algebras
- trac #22921 Shifted tableaux, Krazkiewicz insertion, Haiman mixed insertion
- trac #22922 Faster implementation of LLT polynomials
- Karnofsky–Rhodes and McCammond expansion of a rooted graph
- your wish?
Wishlist

- trac #23560 Benkart-Kang-Kashiwara crystals for super Lie algebras
- trac #22921 Shifted tableaux, Krazkiewicz insertion, Haiman mixed insertion
- trac #22922 Faster implementation of LLT polynomials
- Karnofsky–Rhodes and McCammond expansion of a rooted graph
- your wish?