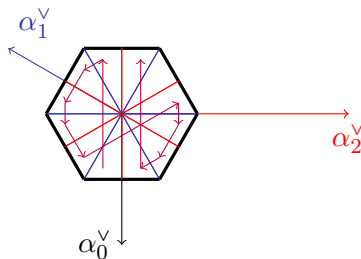


Mathematics and Computers

Anne Schilling, UC Davis

CIRM, February 11, 2019



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!

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!

$$B(\begin{array}{|c|c|} \hline & \\ \hline & \\ \hline \end{array})$$


Lonely Programming ...

Before 2008:

- 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 ...

Before 2008:

- 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 ...

Before 2008:

- 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 ...

Before 2008:

- 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 ...

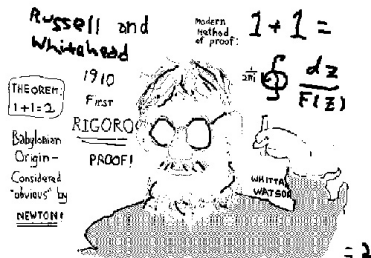
Before 2008:

- 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

What can SageMath do?

```
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)
```

Implementation of a crystal

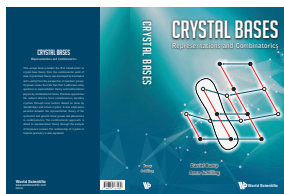
```
class HighestWeightCrystalOfTypeA(UniqueRepresentation, Parent):  
    def __init__(self, n = 3):  
        Parent.__init__(self, category = ClassicalCrystals())  
        self.n = n  
        self._cartan_type = CartanType(['A',n])  
        self.module_generators = [ self(1) ]  
  
    def _repr_(self):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

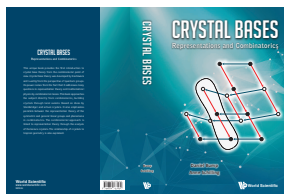
Active tickets!

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

Active tickets!

SageMath

SageMath's mission: Creating a viable free open source alternative to MagmaTM, MapleTM, MathematicaTM, and MatlabTM.

- free
- open source
- anyone can contribute!

Design: SageMath is build around Python (general-purpose programming language)

SageMath

SageMath's mission: Creating a viable free open source alternative to MagmaTM, MapleTM, MathematicaTM, and MatlabTM.

- free
- open source
- anyone can contribute!

Design: SageMath is build around Python (general-purpose programming language)

SageMath

SageMath's mission: Creating a viable free open source alternative to MagmaTM, MapleTM, MathematicaTM, and MatlabTM.

- free
- open source
- anyone can contribute!

Design: SageMath is build around Python (general-purpose programming language)

SageMath

SageMath's mission: Creating a viable free open source alternative to MagmaTM, MapleTM, MathematicaTM, and MatlabTM.

- free
- open source
- anyone can contribute!

Design: SageMath is build around Python (general-purpose programming language)

SageMath

SageMath's mission: Creating a viable free open source alternative to MagmaTM, MapleTM, MathematicaTM, and MatlabTM.

- free
- open source
- anyone can contribute!

Design: SageMath is build around Python (general-purpose programming language)

SageMath

History:

- SageMath project began by William Stein in 2005
SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system SageMath as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- **CoCal** <https://cloud.sagemath.com>
For my classes, I use a **local CoCal** Server
<https://square.math.ucdavis.edu>
- **Install** SageMath <http://sagemath.org>

SageMath

History:

- SageMath project began by William Stein in 2005
SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system SageMath as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- **CoCal** <https://cloud.sagemath.com>
For my classes, I use a **local CoCal** Server
<https://square.math.ucdavis.edu>
- **Install** SageMath <http://sagemath.org>

SageMath

History:

- SageMath project began by William Stein in 2005
SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system SageMath as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- **CoCal** <https://cloud.sagemath.com>
For my classes, I use a **local CoCal** Server
<https://square.math.ucdavis.edu>
- **Install** SageMath <http://sagemath.org>

SageMath

History:

- SageMath project began by William Stein in 2005
SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- `sagenb.org` now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system SageMath as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- **CoCal** <https://cloud.sagemath.com>
For my classes, I use a **local CoCal Server**
<https://square.math.ucdavis.edu>
- **Install SageMath** <http://sagemath.org>

SageMath

History:

- SageMath project began by William Stein in 2005
SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system SageMath as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- **CoCal** <https://cloud.sagemath.com>
For my classes, I use a **local CoCal** Server
<https://square.math.ucdavis.edu>
- **Install** SageMath <http://sagemath.org>

Experimental Mathematics

