# The Coercion Framework of SageMath

April 15, 2015

# 1 Multiplying Apples and Oranges: Transparent Arithmetic with different Data Types in SageMath

## 1.1 *Or:* An Introduction to SageMath's Coercion Framework

**Daniel Krenn**, *TU Graz*, *Austria*

## 1.2 What is this Talk about?

```
In [1]: 15 + 4 / 2015
```

```
Out[1]: 30229/2015
```

```
In [2]: 15.parent()
```

```
Out[2]: Integer Ring
```

```
In [3]: (4 / 2015).parent()
```

```
Out[3]: Rational Field
```

### 1.2.1 What is the Goal of Coercions

from SageMath's documentation:

```
"The primary goal [...] is to be able to transparently do
arithmetic, comparisons, etc. between elements of distinct
sets."
```

### 1.2.2 A short Look behind the Scenes

```
In [4]: QQ.coerce_map_from(ZZ)  # seeing what's going on
```

```
Out[4]: Natural morphism:
          From: Integer Ring
          To:   Rational Field
```

### 1.2.3 An Exercise

```
In [5]: P.<t> = ZZ[]
        P
```

```
Out[5]: Univariate Polynomial Ring in t over Integer Ring
```

```
In [6]: z0 = t
        z0.parent()
```

```
Out[6]: Univariate Polynomial Ring in t over Integer Ring
```

```
In [7]: z1 = t / 2
        z1.parent()
```

```
Out[7]: Univariate Polynomial Ring in t over Rational Field
```

```
In [8]: z2 = t / 1
        z2.parent()
```

```
Out[8]: Univariate Polynomial Ring in t over Rational Field
```

```
In [9]: z3 = 1 / t
        z3.parent()
```

```
Out[9]: Fraction Field of Univariate Polynomial Ring in t over Integer Ring
```

## 1.3 Coercions vs. Conversions

```
In [10]: ZZ(2/1)
```

```
Out[10]: 2
```

```
In [11]: ZZ(3/2)
```

```
        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-11-8c481e7b9db1> in <module>()
    ----> 1 ZZ(Integer(3)/Integer(2))


        /local/data/krenn/sage/6.6/src/sage/structure/parent.pyx in sage.structure.parent.Parent.__call_
       1092         if mor is not None:
       1093             if no_extra_args:
    -> 1094                 return mor._call_(x)
       1095             else:
       1096                 return mor._call_with_args(x, args, kwds)


        /local/data/krenn/sage/6.6/src/sage/rings/rational.pyx in sage.rings.rational.Q_to_Z._call_ (bui
       3872         """
       3873         if not mpz_cmp_si(mpq_denref((<Rational>x).value), 1) == 0:
    -> 3874             raise TypeError, "no conversion of this rational to integer"
       3875         cdef integer.Integer n
       3876         n = <integer.Integer>PY_NEW(integer.Integer)


        TypeError: no conversion of this rational to integer
```

## 1.4 Another Example: Real Numbers

```
In [12]: RR
```

Out[12]: Real Field with 53 bits of precision

```
In [13]: a = RR(pi)   # a conversion occurs here
         a
```

Out[13]: 3.14159265358979

```
In [14]: pi.parent()
```

Out[14]: Symbolic Ring

```
In [15]: R2 = RealField(2)
         R2
```

Out[15]: Real Field with 2 bits of precision

```
In [16]: b = R2(3)
         b
```

Out[16]: 3.0

```
In [17]: c = a + b
         c, c.parent()
```

Out[17]: (6.0, Real Field with 2 bits of precision)

### 1.4.1 Comparisions

```
In [18]: a == b
```

Out[18]: True

## 1.5 A more Challenging Example

```
In [19]: P   # above: P.<t> = ZZ[]
```

Out[19]: Univariate Polynomial Ring in t over Integer Ring

```
In [20]: d = (t^2 + 15*t) + 4/2015
         d
```

Out[20]: t^2 + 15*t + 4/2015

```
In [21]: d.parent()
```

Out[21]: Univariate Polynomial Ring in t over Rational Field

```
In [22]: P.coerce_map_from(QQ) is None, QQ.coerce_map_from(P) is None
```

Out[22]: (True, True)

### 1.6 Looking behind the Scene: Now really...

In [23]:
```
from sage.structure.element import get_coercion_model
cm = get_coercion_model()
cm  # What a strange object!
```

Out[23]:
```
<sage.structure.coerce.CoercionModel_cache_maps object at 0x7f9615fc7050>
```

In [24]:
```
cm.common_parent(P, QQ)
```

Out[24]:
```
Univariate Polynomial Ring in t over Rational Field
```

In [25]:
```
cm.explain(P, QQ)
```

```
Action discovered.
    Right scalar multiplication by Rational Field on Univariate Polynomial Ring in t over Integer Ring
Result lives in Univariate Polynomial Ring in t over Rational Field
```

Out[25]:
```
Univariate Polynomial Ring in t over Rational Field
```

### 1.7 Discovering new Parents

In [26]:
```
M.<m, n> = ZZ[]
M
```

Out[26]:
```
Multivariate Polynomial Ring in m, n over Integer Ring
```

In [27]:
```
alpha = m^2 * n + 42 * n^2
alpha
```

Out[27]:
```
m^2*n + 42*n^2
```

In [28]:
```
N.<n, o> = QQ[]
N
```

Out[28]:
```
Multivariate Polynomial Ring in n, o over Rational Field
```

In [29]:
```
beta = n^2 / 3 + o
beta
```

Out[29]:
```
1/3*n^2 + o
```

In [30]:
```
gamma = alpha + beta
gamma, gamma.parent()
```

Out[30]:
```
(m^2*n + 127/3*n^2 + o,
 Multivariate Polynomial Ring in m, n, o over Rational Field)
```

In [31]:
```
cm.explain(M, N)
```

```
Coercion on left operand via
    Conversion map:
      From: Multivariate Polynomial Ring in m, n over Integer Ring
      To:   Multivariate Polynomial Ring in m, n, o over Rational Field
Coercion on right operand via
    Conversion map:
      From: Multivariate Polynomial Ring in n, o over Rational Field
      To:   Multivariate Polynomial Ring in m, n, o over Rational Field
Arithmetic performed after coercions.
Result lives in Multivariate Polynomial Ring in m, n, o over Rational Field
```

Out[31]:
```
Multivariate Polynomial Ring in m, n, o over Rational Field
```

### 1.7.1 Constructions for Discovering new Parents

```
In [32]: M.construction()
```

```
Out[32]: (MPoly[m,n], Integer Ring)
```

```
In [33]: N.construction()
```

```
Out[33]: (MPoly[n,o], Rational Field)
```

```
In [34]: QQ.construction()
```

```
Out[34]: (FractionField, Integer Ring)
```

```
In [35]: cm.common_parent(M, N).construction()
```

```
Out[35]: (MPoly[m,n,o], Rational Field)
```

## 1.8 Properties of Coercions / Axioms

1. A coercion is defined on all elements of a parent.
2. Coercions are structure preserving.
3. There is at most one coercion from one parent to another.
4. Coercions can be composed.
5. The identity is a coercion