SageMath & Algebra #1

2017年4月16日
Cabri 研究会
生越　茂樹

§ 1. *SageMath* の特徴

- Pythonを基盤としたオープンソースのCAS (初版2005年 by William Stein)
- *Mathematica*, Maple, Magma, MatLAB などと並ぶ事が 長期目標.
- 様々なGNU(オープンソース)のソフトを統合している． Maxima や Python 等に加え GAP, PARI など代数学/数論のソフトも内蔵しているので, 少なくともガロア群の計算などは *Mathematica* より上と思われる.
- GNUのソフトだけでなく，Python を経由し，*Mathematica* なども使える.
  (逆に Mathematica から SageMath を呼び出せる .nb もあるらしい．)
- Linux，Mac　へは　直接インストール可能.（しかしThinkpadX40には重すぎ)
- Windowsへは VirtualBox など仮想環境と共に インストール(5G程度)
- インストールしなくともサーバー経由（無料もあり)でも使える（*SageCloud*)
- *LibreMath* (*KnoppixMath*が devian ベースに変更された和製プロジェクト)
  などを利用して，DVD や USBから起動できる.（DVDは遅いが,USBはOK)
- iOS や Android 用のアプリもある. (SageMathで検索)

§ 2. *SageMath* 初めの一歩

1. SageMathCloud（ https://cloud.sagemath.com/ ）でアカウントを作ってログインし,「NewWorkSheet」をクリックし worksheet を作成. (野良サーバーを使っても良い.)
2. セルに入力後は「Evaluate」をクリック, または「Shift＋Enter」で実行
3. 入力セルの削除は,「BackSpace」キーで消去. 出力の消去は画面上部のコマンドからも可能
4. Object志向でクラスに分かれており, 関数には function と class method がある.
5. 大文字と小文字は区別する. 一般にクラス名は Camel型, 関数は snake型 .
6. ストリングは " " または ' ' (どちらもOK) . コメントは「 #」の後に入力.
7. 数学定数は pi, i , I, e など. もし上書きした場合は reset( ), restore('pi e')などで復元可能.
8. 引数は( ) , リストは [ ], トプルは a,b,c など. 例えば plot(sin(x),(x,0,2*pi)) のように入力.
9. リストは 0番からスタート. List の k番目の要素にアクセスするには List[k] と入力.
10. 「コマンド補完」は Tabキー. Class method では「. 」も忘れずに付ける.
11. Helpは コマンドの後に「?」を入力し Tab (Shift +Enter, Ctrl + Space).「??」はより詳しい.
12. 直前の出力への参照は「_ 」(注意して使わないとエラーになる.)
13. 複数コマンドを同じ行に入力するには「; 」で区切る.
14. 「変数への代入」は出力されない. (例「x=2+3; x 」 で初めて 「5 」と出力される.)
15. 自由変数は定義してから使う. 例えば x とy を使いたいときは var('x y') が必要.
16. 繰り返しは [*実行文* for *変数* in *リスト* (必要なら*条件文*)] の形 (リスト内包!)
例えば [1, 4 ,9] は [k^2 for k in range(3)] と入力する
17. x∈A は「x in A」, 否定は「not」.例えば [k^2 for k in range(5) if not k in [2,4]]→ [1,9,25]

www.sagemath.org からのリンクをたどるとTutorialに行ける. しかし日本語のtutorial http://doc.sagemath.org/pdf/ja/tutorial/tutorial-jp.pdf は見つけ難い. (google した方がずっと速い)
Quick Reference（https://wiki.sagemath.org/quickref) には「基本的な」コマンドがまとまっている.

§ 3. *SageMath* と群論

1. 特に参考にしたもの
Group theory and Sage
http://doc.sagemath.org/html/en/thematic_tutorials/group_theory.html
Galois Groups of number theory
http://doc.sagemath.org/html/en/reference/number_fields/sage/rings/number_field/galois_group.html
Quick Reference for algebra
https://wiki.sagemath.org/quickref?action=AttachFile&do=view&target=quickref-algebra.pdf (添付済)

Sage for Abstract Algebra -Robert Breezer著
(Abstract Algebra –Thomas W Judson著 とセット)

2. Workshop
ウォーミングアップした後, 正三角形の変換群を題材に,
(i)群の定義, 群の同型, 要素の積, Cayley表
(ii) 部分群, 正規部分群, 商群, 商群のCayley表
(iii) 共役群と正規部分群の関係
について遊ぶ. 時間があればGalois群についても少し触れる.

§ リンク集1 - Sageと出会うには

1. Sage 一般
   sagemath (本家)

2. Sage Install
   sagemath (Official site. ここだけで済めば**とても**ラッキー)
   Mathlibre (windows へのインストール. 日本の mathlibreチーム による解説. 図もあり.)
   くろきげんのライブドアBlog (windows へのインストール.裏技も詳しい.)
   小人さんの妄想 (mac, windowsへのインストール)
   さげます (linuxへのインストール方法が3種類.)

3. Sage Online (Server)
   SageMathCloud (ver7使用. 本家. しかし無料会員の場合は, しばしば接続できなくなる)
   SageMath桜 (ver6使用. 個人ユーザーによるサーバー. なお名前は勝手に付けました)
   SageMath立命館 (ver7.0. 立命館大学のサーバー) [サーバーは他にも沢山あるはず.]

4. LiveCD&USB
   mathlibre (ここから .iso ファイル(約4G) をダウンロードし DVDに焼く. その後 DVDから起動し「設定」→「ブータブルUSBの作成」 で, LiveUSB が作れる. 日本のプロジェクト.)
   SageDevianLive (Mathlibreと同様. こちらはsagemathからリンクされている.)

§ リンク集2 - 出会いのあと

1. Official Tutorial
   日本語のtutorialはこちら. 英語の方は sagemathのホームページから沢山リンクされている.

2. Sage一般に関する本
   はじめてのSage –Ted Kosan著, 横田博史 訳
   数論研究者のためのSage -木村巌 著 (基本的な使い方も書いてある. 以上2つは日本語)
   Sage Beginners' Guide – Finch,Craig 著 (手取り足取り.)
   Sage for undergraduates – Gregory V Bard著(読んでないが良さそう)
   Sage for power users -William Stein著 (William Stein は Sage の創設者. ちょっと読んだ)

3. Sage と代数学
   Sage Quickstart for Abstract Algebra , Galois Groups of number theory - Official Tutorial
   Elementary number theory - William Stein著 (数学とsageを 3:1で混ぜた様な本)
   Sage for Abstract Algebra -Robert Breezer著 (下の本とセット. Sageの解説中心. お勧め.)
   Abstract Algebra –Thomas W Judson著 (上の本とセット. 数学的な解説中心)

*英語の本は sagemathのホームページから 他にも沢山(40冊程)見つかる. 全て無料!

4. その他のonline情報
   ペンギンは空を飛ぶ –ガロア群がSageで「サクッと」求まる. これで Sage を使ってみる気になった.
   takemotoさんのページ –非常にまとまった概論. 初めの一歩.

Workshop: Part1. introduction

1. 関数のグラフ

var(' variables')

plot3d(f(x,y), (x,a,b),(y,c,d))

plot(f(x), (x,a,b))　　*(x,a,b)は[x,a,b]でもOK

```
var('x y')     #関数宣言. xとyの間にblankが必要.
plot3d(x^2+y^2,(x,-2,2),(y,-2,2))  #左下のマークまたはグラフの上をClick.拡大縮小は,マウスの中央の輪を使う.()の代わりに[]でもOK.
```

```
plot(log(x)/x,(x,1,100)) # xはすでに関数宣言されているのでここでは不要.
```

2. 方程式

solve(f(x)==0,x)

solve([f(x,y)==0,g(x,y)==0],[x,y])　　*[x,y]は(x,y), x,y (トプル) でもOK

```
solve(2*x^3-7*x^2+10*x-6==0,x)
```

3.微積分

diff(f(x),x) または derivative(f(x),x)

integrate(f(x), x) 不定積分

integrate(f(x),(x,a,b)) 定積分　　*(x,a,b)は[x,a,b]でもOK

```
diff(x^2*sin(x),x)  #derivative もOK.
```

```
int=integrate(x^2*cos(x)+2*x*sin(x),x);int  #不定積分. 微分と違い int() では駄目.
```

```
int.simplify() #最も基本のsimplify. しかし他にも7～8種類の .simplify_foo がある
```

```
int.simplify_full()
```

```
integratate(sin(x)^10,(x,0,pi/2))   #定積分
```

Part2. 正3角形の置換群 D(3), 正方形の置換群D(4)

1. D(3)について「群の定義, 群の同型, 要素の積, Cayley表を作成」

DihedralGroup(n)， SymmetricGroup(n)， AlternativeGroup(n) - 2面体群,対称群,交代群

A.is_isomorphic(B)　AとBが同型か？　　　A.is_abelian()　Aはabel群か?

Gを置換群とするとき G("(1,2,3)")は, トプル(1,2,3)を 巡回置換(1,2,3)に変換する.

G.cayley_table()　群Gの演算表.（names=letters, elements, digits ,or list)

```
tri=Graph([(1,2),(2,3),(3,1)]); tri.plot() #Graph は グラフ理論のクラス
```

```
T=tri.automorphism_group();T    #Tはtriの自己同型群
```

```
T.list()                        #Tの成分表示(Tがリストの時は使えない.)
```

```
D=DihedralGroup(3);D.list()    #Dは 3次の2面体群 (正3角形の置換群)
```

```
S=SymmetricGroup(3);S.list()     #Sは 3次対称群
```

```
S.is_isomorphic(D)   #SはDと同型か? (この場合は S==D でも同じ結果)
```

```
s=S("(1,2,3)");s   #sはσ. 120度の回転. Sの要素として(1,2,3)を定義. S([(1,2,3)](Python)もOK.
```

```
t=S("(2,3)");t     #tはτ. 対称軸に関する折り返し
```

```
ts=s*t;ts        # sとtの合成 (順序に注意)
```

数学のテキストと順序が逆!! [1,2,3](by s)→[2,3,1](by t)→[3,2,1]なので, s*t≡(s と t の合成).

```
ts([1,2,3])    #群の要素:ts による[1,2,3]の像
```

```
st=t*s;st([1,2,3])       #st≠ts が分る
```

```
S.is_abelian()    #SはAbel群(可換群) か?
```

```
[(), s,s^2 , t,  s*t, s^2*t]   #S={(),s,s^2,t,ts,ts^2}となることが分る.
```

```
S.cayley_table(names='elements')   #左の列→上の行の順に積を合成.  names のoptionは ,letters, elements, digits
と list
```

```
S.cayley_table()  #Default はこれ. names='letters'
```

```
S.cayley_table(names=['id','ts^2' ,'s','s^2','t','ts']) #リストを使って自由に名前をつけられる
```

2. D(3)について「部分群，正規部分群，商群，商群のCayley表」

G.subgroups()- Gの部分群全て(複数形).　G.subgroup(list) - list から生成される部分群 (単数)

G.normal_subgroups() -Gの正規部分群全て

G.quotient(N)- Nが正規部分群であるとき，G/N（GのNによる商群)

G.cosets(N),　G.cosets(N,'right')- GのNによる右剰余類，　G.cosets(N,'left')は左剰余類.

len(list), sorted(list) は function (class method でない)．len→lengthの意味.

繰り返しは [実行文　for　変数　in　リスト（必要なら条件文)]　（リスト内包!)

```
subs=S.subgroups();subs   #Sの部分群全てを求める.
```

```
normals=S.normal_subgroups();normals  #Sの正規部分群全てを求める.
```

```
N=normals[1];N.list() #真の正規部分群は {id,sigma, sigma^2} （回転部分群）だけ
```

```
A3=AlternatingGroup(3);A3.list()   #A3は3次の交代群. A3とNは同型.
```

```
Q=S.quotient(N);Q   # Q=S/N（SのNによる商群)
```

```
Q.list()   #Qは折り返しの群
```

```
Q.cayley_table(names='elements')   #Q(商群)のケイレー表(演算表)
```

以下はQの演算表(cayley table)の検証

```
a=S("(1,2)");a   #「トプル a」を Sの要素 と見る
```

```
Na=[a*S(N[i]) for i in range(3)];Na   #Na. S(N[i])で,N[i]をSの要素に変換.
```

```
aN=[ S(N[i])*a for i in range(3)];aN   #aN
```

```
R=S.cosets(N);R   #SのNによる右剰余類. これを使うとNaはすぐ求まる.
```

```
A=R[1];A      #A=Na=aN=R[1]と分る
```

```
AA=[];junk=[AA.append(S(A[i])*S(A[j])) for i in range(3) for j in range(3) if not S(A[i])*S(A[j]) in AA ];
AA           #A*Aの計算. これに対するコマンドはない.
```

```
sorted(AA)==sorted(N)  ##A*A=N の検証. A*A==N だと False となる.
```

```
AN=[];junk=[AN.append(S(N[i])*S(A[j])) for i in range(3) for j in range(3) if not S(N[i])*S(A[j]) in AN ];AN
```

```
sorted(AN)==sorted(A) #AN=A の検証
```

```
NA=[];junk=[NA.append(S(A[i])*S(N[j])) for i in range(3) for j in range(3) if not S(A[i])*S(N[j]) in NA ]; NA
```

```
sorted(NA)==sorted(A)   #NA=A の検証
```

```
NN=[];junk=[NN.append(S(N[i])*S(N[j])) for i in range(3) for j in range(3) if not S(N[i])*S(N[j]) in NN ]; NN
```

```
sorted(NN)==sorted(N)    #NN=N の検証
```

```
Q.cayley_table(['N','A'])
```

以上から 商群Q=G/N の演算表が確認された． G→Qの準同型写像を作る事もできる．

3. Dの共役類と正規部分群の関係

D.conjugacy_classes_representatives - Dの共役類の代表（適当に選ばれる）

「def func(x): .... return 」- Python関数の定義． 改行とTab に注意

```
reset();                                # reset (本当は不必要ですが,念のため)
square=Graph([(1,2),(2,3),(3,4),(4,1)])  # Graph は グラフ理論のクラス
square.plot()
```

```
A=square.automorphism_group() ;A.list()
```

```
D=DihedralGroup(4);D.list()
```

```
A.is_isomorphic(D)   #AとDは 正方形の自己同型写像
```

```
D.order()    #Dのorder(要素の数)
```

```
reps=D.conjugacy_classes_representatives();reps   #repsは 共役類の「代表」
```

x=(2,4) を代表とするDの共役類; g^(-1) x g （∀g ∈D） の作成

```
x=D("(2,4)");
conj=[];
junks=[conj.append(g^(-1)*x*g) for g in D if not g^(-1)*x*g in conj]
conj
```

上の操作をPython関数にする（indentに注意． lambda関数も使える）

```
def myconj(x):    #「:」を忘れないこと.
    conj=[]       #以下の3行は全てインデントが必要.(indent=1Tab=4*blank)
    junks=[conj.append(g^(-1)*x*g) for g in D if not g^(-1)*x*g in conj]
    return conj
```

```
myconj(x)  #xを代表とする共役類. 以前の結果と一致するか?
```

```
cons=[myconj(rep) for rep in reps];cons  # Dの共役類による分類
```

Dの共役類(conjugacy class)は 5クラスあり，上から順に

恒等変換(１つ)，

対角線に関する対称移動（２つ），

対辺の中点を結ぶ直線に関する対称移動（２つ）

±90度の回転（２つ）

中心に対する対称移動（１つ） 合計8個=order(D)

「これらの共役類の幾つかの和となっている部分群」が「正規部分群」となる.

(∵NがGの正規部分群 ⇔ ∀g∈Gに対し g^(-1)Ng=N ⇔ g^(-1)Ng⊆N)

```
c0,c1,c2,c3,c4=cons;  #tuple unpacking. c0～c4にまとめて代入
```

```
c1   #(対角線に関する対称移動)
```

```
g1=D.subgroup(c1);g1.list()  #g1は「c1から生成されるDの部分群」
```

```
sorted(c0+c1+c4)==sorted(g1)
```

```
g1.is_normal()          #g1=c0+c1+c4(共役類の和)なので g1 は正規部分群
```

```
c2   #(対辺の中点を結ぶ直線に関する対称移動)
```

```
g2=D.subgroup(c2);g2.list()
```

```
sorted(g2)==sorted(c0+c2+c4)
```

```
g2.is_normal(D)      #g2=c0+c2+c4(共役類の和)なので g2 は正規部分群
```

ここでDの全ての正規部分群を求めて，比較してみる.

```
normals=D.normal_subgroups();
[normals[i].list() for i in range(len(normals))]
```

上とc0～c4を比べると，D(4)の正規部分群は上から順に

c0, c0+c4, c0+c2+c4(=g2), c0+c1+c4(=g1), c0+c3+c4, c0+c1+c3+c4(=D)

となっていることが分る. D(3)に関しても同様にできるが省略.

```

```

Sage Quick Reference: Abstract Algebra

B. Balof, T. W. Judson, D. Perkinson, R. Potluri

version 1.0, Sage Version 5.0.1

latest version: `http://wiki.sagemath.org/quickref`

Based on work by P. Jipsen, W. Stein, R. Beezer

Basic Help

com⟨`tab`⟩ complete *command*

`a.`⟨`tab`⟩ all methods for object `a`

`<command>?` for summary and examples

`<command>??` for complete source code

`*`*foo*`*?` list all commands containing *foo*

`_` underscore gives the previous output

`www.sagemath.org/doc/reference` online reference

`www.sagemath.org/doc/tutorial` online tutorial

`load foo.sage` load commands from the file `foo.sage`

`attach foo.sage`
  loads changes to `foo.sage` automatically

Lists

`L = [2,17,3,17]` an ordered list

`L[i]` the ith element of L

Note: lists begin with the 0th element

`L.append(x)` adds x to L

`L.remove(x)` removes x from L

`L[i:j]` the i-th through $(j-1)$-th element of L

`range(a)` list of integers from 0 to $a-1$

`range(a,b)` list of integers from a to $b-1$

`[a..b]` list of integers from a to b

`range(a,b,c)`
  every c-th integer starting at a and less than b

`len(L)` length of L

`M = [i^2 for i in range(13)]`
  list of squares of integers 0 through 12

`N = [i^2 for i in range(13) if is_prime(i)]`
  list of squares of prime integers between 0 and 12

`M + N` the concatenation of lists `M` and `N`

`sorted(L)` a sorted version of `L` (`L` is not changed)

`L.sort()` sorts `L` (`L` is changed)

`set(L)` an unordered list of unique elements

Programming Examples

Print the squares of the integers $0,\ldots,14$:

```
for i in range(15):
    print i^2
```

Print the squares of those integers in $\{0,\ldots,14\}$ that are relatively prime to 15:

```
for i in range(13):
    if gcd(i,15)==1:
       print i^2
```

Preliminary Operations

`a = 3; b = 14`

`gcd(a,b)` greatest common divisor a,b

`xgcd(a,b)`
  triple (d,s,t) where $d=sa+tb$ and $d=\gcd(a,b)$

`next_prime(a)` next prime after a

`previous_prime(a)` prime before a

`prime_range(a,b)` primes p such that $a \le p < b$

`is_prime(a)` is a prime?

`b % a` the remainder of b upon division by a

`a.divides(b)` does a divide b?

Group Constructions

Permutation multiplication is left-to-right.

`G = PermutationGroup([[(1,2,3),(4,5)],[(3,4)]])`
  perm. group with generators $(1,2,3)(4,5)$ and $(3,4)$

`G = PermutationGroup(["(1,2,3)(4,5)","(3,4)"])`
  alternative syntax for defining a permutation group

`S = SymmetricGroup(4)` the symmetric group, S_4

`A = AlternatingGroup(4)` alternating group, A_4

`D = DihedralGroup(5)` dihedral group of order 10

`Ab = AbelianGroup([0,2,6])` the group $\mathbb{Z}\times\mathbb{Z}_2\times\mathbb{Z}_6$

`Ab.0, Ab.1, Ab.2` the generators of `Ab`

`a,b,c = Ab.gens()`
  shorthand for `a = Ab.0; b = Ab.1; c = Ab.2`

`C = CyclicPermutationGroup(5)`

`Integers(8)` the group $\mathbb{Z}_8$

`GL(3,QQ)` general linear group of 3×3 matrices

`m = matrix(QQ,[[1,2],[3,4]])`

`n = matrix(QQ,[[0,1],[1,0]])`

`MatrixGroup([m,n])`
  the (infinite) matrix group with generators `m` and `n`

`u = S([(1,2),(3,4)]); v = S((2,3,4))` elements of `S`

`S.subgroup([u,v])`
  the subgroup of `S` generated by `u` and `v`

`S.quotient(A)` the quotient group `S/A`

`A.cartesian_product(D)` the group `A×D`

`A.intersection(D)` the intersection of groups `A` and `D`

`D.conjugate(v)` the group $\mathtt{v}^{-1}\mathtt{Dv}$

`S.sylow_subgroup(2)` a Sylow 2-subgroup of `S`

`D.center()` the center of `D`

`S.centralizer(u)` the centralizer of `x` in `S`

`S.centralizer(D)` the centralizer of `D` in `S`

`S.normalizer(u)` the normalizer of `x` in `S`

`S.normalizer(D)` the normalizer of `D` in `S`

`S.stabilizer(3)` subgroup of `S` fixing 3

Group Operations

`S = SymmetricGroup(4); A = AlternatingGroup(4)`

`S.order()` the number of elements of `S`

`S.gens()` generators of `S`

`S.list()` the elements of `S`

`S.random_element()` a random element of `S`

`u*v` the product of elements `u` and `v` of `S`

`v^(-1)*u^3*v` the element $\mathtt{v}^{-1}\mathtt{u}^3\mathtt{v}$ of `S`

`u.order()` the order of `u`

`S.subgroups()` the subgroups of `S`

`S.normal_subgroups()` the normal subgroups of `S`

`A.cayley_table()` the multiplication table for `A`

`u in S` is `u` an element of `S`?

`u.word_problem(S.gens())`
  write `u` as a product of the generators of `S`

`A.is_abelian()` is `A` abelian?

`A.is_cyclic()` is `A` cyclic?

`A.is_simple()` is `A` simple?

`A.is_transitive()` is `A` transitive?

`A.is_subgroup(S)` is `A` a subgroup of `S`?

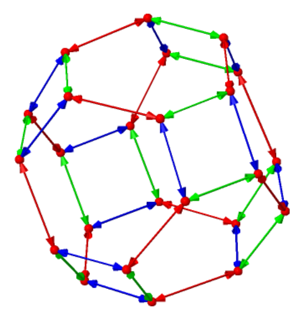
`A.is_normal(S)` is `A` a normal subgroup of `S`?

`S.cosets(A)` the right cosets of `A` in `S`

`S.cosets(A,'left')` the left cosets of `A` in `S`

`g = S.cayley_graph()` Cayley graph of `S`

`g.show3d(color_by_label=True, edge_size=0.01,`
  `vertex_size=0.03)` see below:



Ring and Field Constructions

`ZZ` integral domain of integers, $\mathbb{Z}$
`Integers(7)` ring of integers mod 7, $\mathbb{Z}_7$
`QQ` field of rational numbers, $\mathbb{Q}$
`RR` field of real numbers, $\mathbb{R}$
`CC` field of complex numbers, $\mathbb{C}$
`RDF` real double field, inexact
`CDF` complex double field, inexact
`RR` 53-bit reals, inexact, not same as `RDF`
`RealField(400)` 400-bit reals, inexact
`ComplexField(400)` complexes, too
`ZZ[I]` the ring of Gaussian integers
`QuadraticField(7)` the quadratic field, $\mathbb{Q}(\sqrt{7})$
`CyclotomicField(7)`
smallest field containing $\mathbb{Q}$ and the zeros of $x^7 - 1$
`AA, QQbar` field of algebraic numbers, $\overline{\mathbb{Q}}$
`FiniteField(7)` the field $\mathbb{Z}_7$
`F.<a> = FiniteField(7^3)`
finite field in a of size 7^3, $\mathrm{GF}(7^3)$
`SR` ring of symbolic expressions
`M.<a>=QQ[sqrt(3)]` the field $\mathbb{Q}[\sqrt{3}]$, with $a = \sqrt{3}$.
`A.<a,b>=QQ[sqrt(3),sqrt(5)]`
the field $\mathbb{Q}[\sqrt{3}, \sqrt{5}]$ with $a = \sqrt{3}$ and $b = \sqrt{5}$.
`z = polygen(QQ,'z'); K = NumberField(x^2 - 2,'s')`
the number field in s with defining polynomial $x^2 - 2$
`s = K.0` set `s` equal to the generator of `K`
`D = ZZ[sqrt(3)]`
`D.fraction_field()`
field of fractions for the integral domain `D`

Ring Operations

Note: Operations may depend on the ring

`A = ZZ[I]; D = ZZ[sqrt(3)]` some rings
`A.is_ring()` is A a ring?
`A.is_field()` is A a field?
`A.is_commutative()` is A commutative?
`A.is_integral_domain()`
`True` is A an integral domain?
`A.is_finite()` is A is finite?
`A.is_subring(D)` is A a subring of D?
`A.order()` the number of elements of A
`A.characteristic()` the characteristic of A
`A.zero()` the additive identity of A
`A.one()` the multiplicative identity of A
`A.is_exact()`
`False` if `A` uses a floating point representation
`a, b = D.gens(); r = a + b`
`r.parent()` the parent ring of r (in this case, `D`)
`r.is_unit()` is r a unit?

Polynomials

`R.<x> = ZZ[ ]` `R` is the polynomial ring $\mathbb{Z}[x]$
`R.<x> = QQ[ ]; R = PolynomialRing(QQ,'x'); R = QQ['x']`
`R` is the polynomial ring $\mathbb{Q}[x]$
`S.<z> = Integers(8)[ ]` `S` is the polynomial ring $\mathbb{Z}_8[z]$
`S.<s, t> = QQ[ ]` `S` is the polynomial ring $\mathbb{Q}[s,t]$
`p = 4*x^3 + 8*x^2 - 20*x - 24`
a polynomial in `R` ($= \mathbb{Q}[x]$)
`p.is_irreducible()` is p irreducible over $\mathbb{Q}[x]$?
`q = p.factor()` factor p
`q.expand()` expand `q`
`p.subs(x=3)` evaluates p at $x = 3$
`R.ideal(p)` the ideal in R generated by p
`R.cyclotomic_polynomial(7)`
the cyclotomic polynomial $x^6+x^5+x^4+x^3+x^2+x+1$
`q = x^2-1`
`p.divides(q)` does p divide q?
`p.quo_rem(q)`
the quotient and remainder of `p` upon division by `q`
`gcd(p, q)` the greatest common divisor of p and q
`p.xgcd(q)` the extended gcd of p and q
`I = S.ideal([s*t+2,s^3-t^2])`
the ideal $(st+2, s^3-t^2)$ in `S` ($= \mathbb{Q}[s,t]$)
`S.quotient(I)` the quotient ring, S/I

Field Operations

`A.<a,b>=QQ[sqrt(3),sqrt(5)]`
`C.<c> = A.absolute_field()`
"flattens" a relative field extension
`A.relative_degree()`
the degree of the relative extension field
`A.absolute_degree()`
the degree of the absolute extension
$\mathrm{r} = \mathrm{a} + \mathrm{b}$; `r.minpoly()`
the minimal polynomial of the field element `r`
`C.is_galois()` is `C` a Galois extension of Q?