The SAGE Coercion Model

Robert Bradshaw

Oct 2, 2007

Outline of Topics

Examples

Ring Examples from SD 4

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not Frac($\mathbf{Z}[x]$))

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not $\mathrm{Frac}(\mathbf{Z}[x])$)
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not $\mathrm{Frac}(\mathbf{Z}[x])$)
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not Frac($\mathbf{Z}[x]$))
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.
- $\mathbf{Z}[x] + \mathbf{Z}[y]$ error due to unknown relation between x and y and ambiguous order

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not $\mathrm{Frac}(\mathbf{Z}[x])$)
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.
- $\mathbf{Z}[x] + \mathbf{Z}[y]$ error due to unknown relation between x and y and ambiguous order
- $\mathbf{Q}[\zeta_m] + \mathbf{Q}[\zeta_n] \in \mathbf{Q}[\zeta_{lcm(m,n)}]$ as cyclotomic fields are created *with* an embedding into $\bar{\mathbf{Q}}$

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not $\mathrm{Frac}(\mathbf{Z}[x])$)
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.
- $\mathbf{Z}[x] + \mathbf{Z}[y]$ error due to unknown relation between x and y and ambiguous order
- $\mathbf{Q}[\zeta_m] + \mathbf{Q}[\zeta_n] \in \mathbf{Q}[\zeta_{lcm(m,n)}]$ as cyclotomic fields are created *with* an embedding into $\bar{\mathbf{Q}}$
- $\mathbf{F}_{p^n} + \mathbf{F}_{p^m}$ works using Conway polynomials

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not $\mathrm{Frac}(\mathbf{Z}[x])$)
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.
- $\mathbf{Z}[x] + \mathbf{Z}[y]$ error due to unknown relation between x and y and ambiguous order
- $\mathbf{Q}[\zeta_m] + \mathbf{Q}[\zeta_n] \in \mathbf{Q}[\zeta_{lcm(m,n)}]$ as cyclotomic fields are created *with* an embedding into $\bar{\mathbf{Q}}$
- $\mathbf{F}_{p^n} + \mathbf{F}_{p^m}$ works using Conway polynomials
- $\mathbf{Z}/16\mathbf{Z} + \mathbf{Z}/12\mathbf{Z} \in \mathbf{Z}/4\mathbf{Z}$

Examples

Ring Examples from SD 4

- $\mathbf{Z}[x]/\mathbf{Z} \in \mathbf{Q}[x]$ (not Frac($\mathbf{Z}[x]$))
- $\mathbf{Q} + \mathbf{Z}[x] \in \mathbf{Q}[x]$ and $\mathbf{Z}/5\mathbf{Z} + \mathbf{Z}[x] \in \mathbf{Z}/5\mathbf{Z}[x]$
- $\mathbf{Q} * \mathbf{Z}/5\mathbf{Z}$ error due to no morphism from "all" of $\mathbf{Q}$ into $\mathbf{Z}/5\mathbf{Z}$.
- $\mathbf{Z}[x] + \mathbf{Z}[y]$ error due to unknown relation between x and y and ambiguous order
- $\mathbf{Q}[\zeta_m] + \mathbf{Q}[\zeta_n] \in \mathbf{Q}[\zeta_{lcm(m,n)}]$ as cyclotomic fields are created *with* an embedding into $\bar{\mathbf{Q}}$
- $\mathbf{F}_{p^n} + \mathbf{F}_{p^m}$ works using Conway polynomials
- $\mathbf{Z}/16\mathbf{Z} + \mathbf{Z}/12\mathbf{Z} \in \mathbf{Z}/4\mathbf{Z}$
- $\mathbf{Q} + \mathrm{Matrix}_{n,m}(\mathbf{Z}[x]) \in \mathrm{Matrix}_{n,m}(\mathbf{Q}[x])$

Cases

Three main cases for arithmetic between R, S.

- (Easy) $R \to S$ or $S \to R$ canonically

Cases

Three main cases for arithmetic between R, S.

- (Easy) $R \to S$ or $S \to R$ canonically
- (Medium) $R \to S$ or $S \to R$ with specified embedding
    - $\mathbb{Q}[\zeta_n] \to \mathbb{C}$ for `CyclotomicField`
    - Quotient objects

Cases

Three main cases for arithmetic between R, S.

- (Easy) $R \to S$ or $S \to R$ canonically
- (Medium) $R \to S$ or $S \to R$ with specified embedding
  - Embedding may be implicit
  - $\mathbb{Q}[\zeta_n] \to \mathbb{C}$ for `CyclotomicField`
  - Quotient objects

Cases

Three main cases for arithmetic between R, S.

- (Easy) $R \to S$ or $S \to R$ canonically
- (Medium) $R \to S$ or $S \to R$ with specified embedding
  - Embedding may be implicit
  - $\mathbb{Q}[\zeta_n] \to \mathbb{C}$ for `CyclotomicField`
  - Quotient objects
- (Hard) $R \to Z$ and $S \to Z$ canonically ("pushouts")

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$
  - $Mat_{2\times 2}(\mathbb{Q}_p)$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$
  - $Mat_{2\times 2}(\mathbb{Q}_p)$

Definition

Let a **construction tower** of R be a series of objects R_i and functors F_i such that $R_{i+1} = F_i(R_i)$ with $R_n = R$.

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$
  - $Mat_{2\times 2}(\mathbb{Q}_p)$

Construction Functors and Towers

- Most objects are the result of applying functors to simpler objects
  - $\mathbb{Z}_p[x, y]$
  - $\mathbb{Q}(\sqrt{5})[[t]]$
  - $Mat_{2\times 2}(\mathbb{Q}_p)$

Definition

Let a **construction tower** of R be a series of objects R_i and functors F_i such that $R_{i+1} = F_i(R_i)$ with $R_n = R$.

Pushforwards

Pushforwards need a reasonable initial object

Pushforwards

Pushforwards need a reasonable initial object

- Case 1. Construction towers converge

Pushforwards

Pushforwards need a reasonable initial object

- Case 1. Construction towers converge
- Case 2. Canonical coercion between towers

Merging Towers

(example)

Implementation

(demo)

Requirements

Implementation in SAGE

- Fast

Requirements

Implementation in SAGE

- Fast
- Non-intrusive

Requirements

Implementation in SAGE

- Fast
- Non-intrusive
- Transitive

Requirements

Implementation in SAGE

- Fast
- Non-intrusive
- Transitive
- Correct

Implementation

`sage: a + b`

Executes

```
if have_same_parent(a,b):
    return a.add_c(b)
else: # bin_op_c
    if A = lookup_action(a,b, op):
        return A(a,b)
    if xmap, ymap = lookup_coercion(a,b)
        return xmap(a) + ymap(b)
    Fail
```

Everything is cached for speed (custom dict)

Special functions

- `R.coerce_map_from()`
- `R.get_action()`
- `cannonical_coercion`
- `pushout`

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad
- Contexts `with Embeddings(...)`

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad
- Contexts `with Embeddings(...)`
- At ring creation time `R = Ring(..., embeddings=X)`

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad
- Contexts `with Embeddings(...)`
- At ring creation time `R = Ring(..., embeddings=X)`
- Wrappers `R = AssertEmbeddings(T, S, f)` where $f : T \to S$

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad
- Contexts `with Embeddings(...)`
- At ring creation time `R = Ring(..., embeddings=X)`
- Wrappers `R = AssertEmbeddings(T, S, f)` where $f : T \to S$

Currently needs some cleaning up.

User embedding (proposal)

User embedding (proposal)

- Mutable parents and side effects can be bad
- Contexts `with Embeddings(...)`
- At ring creation time `R = Ring(..., embeddings=X)`
- Wrappers `R = AssertEmbeddings(T, S, f)` where $f : T \to S$

Currently needs some cleaning up.
What do you,as a developer, want to define?