

A FAIR File Format for Mathematical Software

Antony Della Vecchia

Joint work with M. Joswig and B. Lorenz

TU Berlin

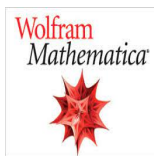
ICMS Mathematical Research Data 2024-07-24



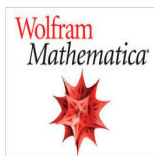
The Importance of Files for Computer Algebra



- People often have a preferred software system.



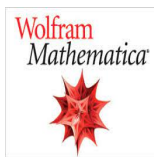
The Importance of Files for Computer Algebra



- People often have a preferred software system.
- Computations can be expensive.



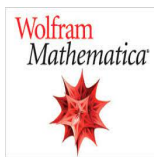
The Importance of Files for Computer Algebra



- People often have a preferred software system.
- Computations can be expensive.
- Software changes often and requires maintenance.



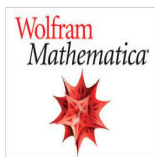
The Importance of Files for Computer Algebra



- People often have a preferred software system.
- Computations can be expensive.
- Software changes often and requires maintenance.
- Data is often more valuable than the software.



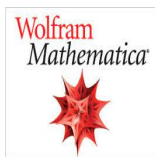
The Importance of Files for Computer Algebra



- People often have a preferred software system.
- Computations can be expensive.
- Software changes often and requires maintenance.
- Data is often more valuable than the software.
- Verification of results is at most as computationally expensive.



The Importance of Files for Computer Algebra



- People often have a preferred software system.
- Computations can be expensive.
- Software changes often and requires maintenance.
- Data is often more valuable than the software.
- Verification of results is at most as computationally expensive.



History of File Formats

```
* Class:      LP
* Rows:       5
* Columns:    2
* Format:     MPS
*
Name          unnamed#0
ROWS
  N  C0000000
  ...
  G  R0000003
COLUMNS
  x1  C0000000  2      R0000000  1
  x1  R0000001 -1      R0000002  1
  x2  C0000000  3      R0000002  1
  x2  R0000003 -1
RHS
  B  C0000000 -1      R0000000 -1
  B  R0000001 -1      R0000002 -1
  B  R0000003 -1
BOUNDS
  FR BND      x1
  FR BND      x2
ENDATA
```

- The LP file format and the MPS file format. IBM [1970s] (industry standards)
- Mathematica Notebooks. Wolfram Mathematica [1988]
- OpenMath (tree structure). Mike Dewar [2000]
- IPython 0.12 Interactive Browser Notebooks (Jupyter) [2011]
- polymake File Format. E. Gawrilow, S. Hampe, and M. Joswig [2016]



Storing Computer Algebra Data

- It's common to have multiple perspectives on an object in mathematics.
- While storing mathematical data a choice of perspective must be made.
- Such a choice might not be describeable in an email.

Say we want to store:

$$p = 2y^3z^4 + (\mathbf{a} + 3)z^2 + 5\mathbf{a}y + 1$$



Storing Computer Algebra Data

- It's common to have multiple perspectives on an object in mathematics.
- While storing mathematical data a choice of perspective must be made.
- Such a choice might not be describeable in an email.

Say we want to store:

$$p = 2y^3z^4 + (\mathbf{a} + 3)z^2 + 5\mathbf{a}y + 1$$

- Some technicalities with the coefficients.
- Is y considered a coefficient of z ?
- What is \mathbf{a} ?
- How can we guarantee the objects behave as expected on load?



The `mrdf` File Format

- JSON file format.
- Similar to `polymake` format but generalizes to include algebraic data.
- Namespaces for semantic separation.
- References stored with UUIDs.
- Prototype developed using `OSCAR`.



Demo

```
julia> using Oscar

julia> F = GF(7)
Prime field of characteristic 7

julia> L, a = finite_field(x^2 + 1)
(Finite field of degree 2 and characteristic 7, o)

julia> Lyz, (y, z) = L[:, :z]
(Multivariate polynomial ring in 2 variables over L, FqMPolyRingElem[y, z])

julia> p = 2 * z^4 * y^3 + (a + 3) * z^2 + 5 * a * y + 1
2*y^3*z^4 + 5*o*y + (o + 3)*z^2 + 1

julia> q = z^2 + 3 * y
3*y + z^2

julia> save("p.mrdi", p)

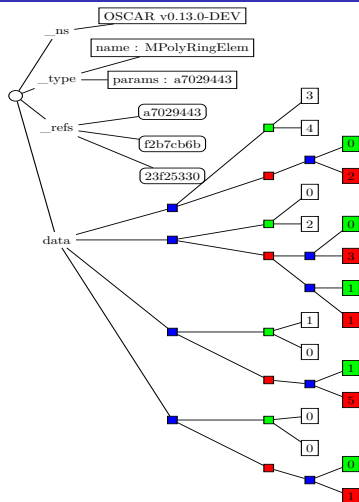
julia> save("q.mrdi", q)

julia> Oscar.reset_global_serializer_state()
Dict{Base.UUID, Any}{}

julia> load("p.mrdi") * load("q.mrdi")
6*y^4*z^4 + 2*y^3*z^6 + o*y^2 + (o + 2)*y*z^2 + 3*y + (o + 3)*z^4 + z^2
```



Tree Structure



$$2y^3z^4$$

$$(a+3)z^2$$

$$5ay$$

$$1$$



Example File Serialized with OSCAR

```
{
  "_ns": { "Oscar": [ "https://github.com/oscar-system/Oscar.jl", "1.0.0" ] },
  "_type": {
    "name": "MPolyRingElem",
    "params": "869a359a-43d3-43f4-9821-0af9346be019"
  },
  "data": [[["3", "4"], [{"0", "2"}]],
    [{"0", "2"}, [{"0", "3"}, {"1", "1"}]],
    [{"1", "0"}, [{"1", "5"}]],
    [{"0", "0"}, [{"0", "1"}]],
  "_refs": {
    "152ac7bd-e85a-4b36-acc2-743ade2cad4f": {
      "data": { "base_ring": { "data": "7", "_type": "FqField"},
        "symbols": ["x"] },
      "_type": "PolyRing"
    },
    "869a359a-43d3-43f4-9821-0af9346be019": {
      "data": {
        "base_ring": "a8309b96-caec-443c-bedb-e23bb0634c14",
        "symbols": [ "y", "z" ]
      },
      "_type": "MPolyRing" },
    "a8309b96-caec-443c-bedb-e23bb0634c14": {
      "data": {
        "def_pol": {
          "data": [{"0", "1"}, {"2", "1"}],
          "_type": {
            "name": "PolyRingElem",
            "params": "152ac7bd-e85a-4b36-acc2-743ade2cad4f"
          }
        }
      },
      "_type": "FqField"
    }
  }
}
```



Namespaces

- We make no attempt to formalise the language of Mathematical Research Data.
- We rely on namespaces to pin the semantics to a specific software.
- Open source software provides example implementations for saving and loading.
- Freedom to upgrade encoding.
- Similar with PDF.



Serialized Toric Divisor

```
{
  "_ns": { ...
  "_type": { "name": "ToricDivisorClass", "params": "b66b610c-f21b-4b3d-8b1f-267e8e65ec7" },
  "data": [ "5", "0", "0" ],
  "_refs": {
    "b66b610c-f21b-4b3d-8b1f-267e8e65ec7a": {
      "_type": "NormalToricVariety",
      "data": {
        "_ns": { "polymake": [ "https://polymake.org", "4.12" ] },
        "_type": "fulton::NormalToricVariety",
        "FAN_AMBIENT_DIM": 2,
        "PROJECTIVE": true,
        "MAXIMAL_CONES": [ ... ],
        "PSEUDO_REGULAR": true,
        "COMPLETE": true,
        "DIVISOR": [ ... ],
        "RAYS": [ [ "1", "0" ], [ "0", "1" ], [ "-1", "-1" ] ],
        ...
      }
    }
  }
}
```



Schema



- A schema defines a structure for data.

Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>



Schema



Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>

- A schema defines a structure for data.
- Schema languages. (RELAX NG [2002], JSON Schema [2022])



Schema



Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>

- A schema defines a structure for data.
- Schema languages. (RELAX NG [2002], JSON Schema [2022])
- Is possible to define recursive structure.



Schema



Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>

- A schema defines a structure for data.
- Schema languages. (RELAX NG [2002], JSON Schema [2022])
- Is possible to define recursive structure.
- Schemata allow data to be validated before loading.



Schema



Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>

- A schema defines a structure for data.
- Schema languages. (RELAX NG [2002], JSON Schema [2022])
- Is possible to define recursive structure.
- Schemata allow data to be validated before loading.
- Adds structure to document based databases.



Schema



Figure: <https://www.pexels.com/photo/plastic-shape-sorter-toy-11030155/>

- A schema defines a structure for data.
- Schema languages. (RELAX NG [2002], JSON Schema [2022])
- Is possible to define recursive structure.
- Schemata allow data to be validated before loading.
- Adds structure to document based databases.
- PolyDB, Paffenholz [2017]



Data Collections

- Johnson Solids.
- Non-general type surfaces in \mathbb{P}^4 .
- F-Theory models.
- Small Phylogenetic Trees. (Luis David Garcia-Puente, Jacob Porter)



Thank You!

