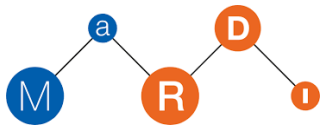
A FAIR File Format

Antony Della Vecchia

Technische Universität Berlin

2023-03-14



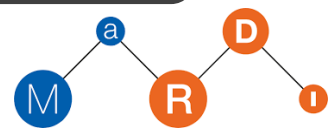
The MaRDI Computer Algebra Team

TU Berlin

- Antony Della Vecchia – data formats
- Michael Joswig
- Lars Kastner – guidelines for reproducibility

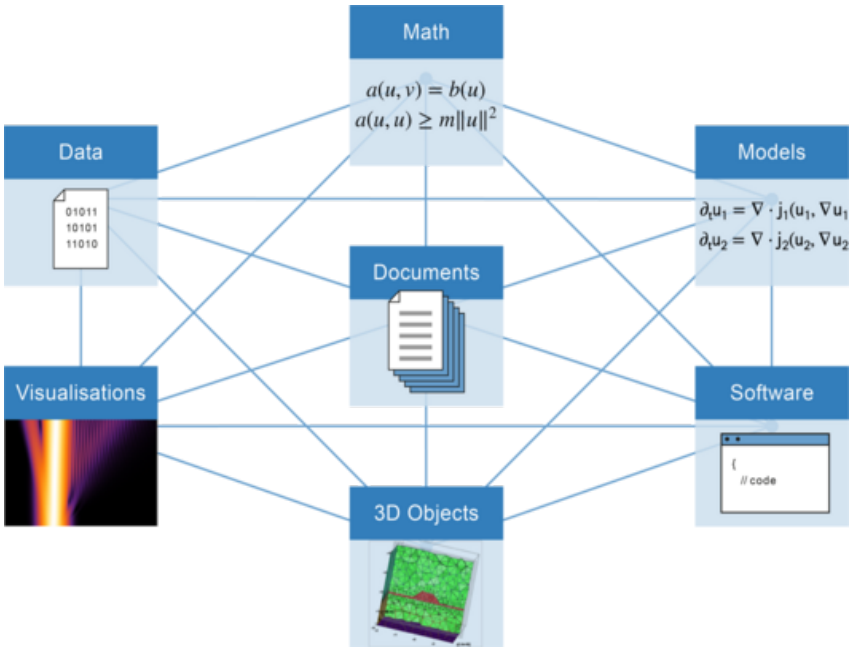
TU Kaiserslautern

- Wolfram Decker
- Claus Fieker
- Jeroen Hanselman – technical peer review
- Max Horn



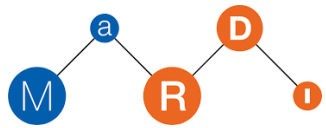
Building an Infrastructure

- Develop a mathematical research data infrastructure.
- Set standards for confirmable workflows and certified mathematical research data.
- Provide services to both the mathematical and wider scientific community.



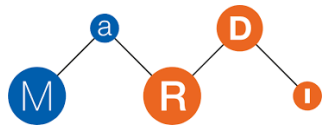
Why Files?

- People often have a preferred software system.



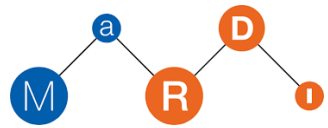
Why Files?

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



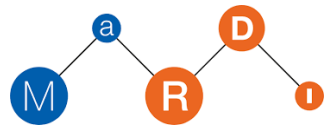
Why Files?

- People often have a preferred software system.
- Computations can be expensive.
- Verification of results is at most as computationally expensive.



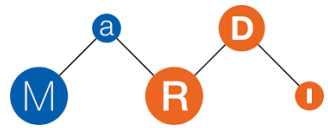
Why Files?

- People often have a preferred software system.
- Computations can be expensive.
- Verification of results is at most as computationally expensive.
- Software changes often.



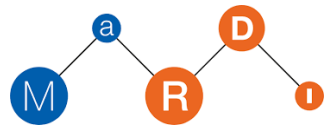
Storing Mathematical Data

- It's common to have multiple perspectives on an object in mathematics.



Storing Mathematical Data

- It's common to have multiple perspectives on an object in mathematics.
- While storing mathematical data a choice of perspective must me made.

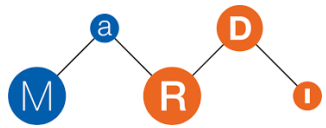


Storing Mathematical Data

- It's common to have multiple perspectives on an object in mathematics.
- While storing mathematical data a choice of perspective must me made.

Say we want to store:

$$p(x, y, z) = xy - z^2$$



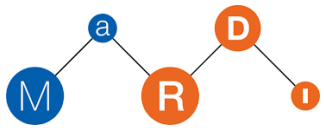
Storing Mathematical Data

- It's common to have multiple perspectives on an object in mathematics.
- While storing mathematical data a choice of perspective must me made.

Say we want to store:

$$p(x, y, z) = xy - z^2$$

- Some technicalities with the coefficients.
- Is x considered as a coefficient of y?

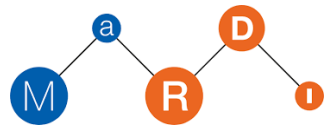


Storing LPs

```
MINIMIZE
  obj: +2 x1 +3 x2 +1
Subject To
  ie0: +1 x1 >= -1
  ie1: -1 x1 >= -1
  ie2: +1 x2 >= -1
  ie3: -1 x2 >= -1
BOUNDS
  x1 free
  x2 free
END
```

```
* 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
    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
```

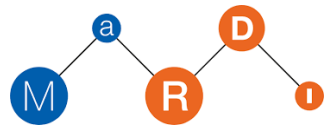
- LPs heavily used in industry.
- Industry natural define standards.
- The LP file format, the MPS file format.



The Polymake File Format

```
<?xml version="1.0" encoding="utf-8"?>
<?pm chk="56e977e8"?>
<object name="square" type="polytope::Polytope&lt;Rational&gt;"
        version="3.0"
        xmlns="http://www.math.tu-berlin.de/polymake/#3">
  <description><![CDATA[cube of dimension 2]]></description>
  <property name="VERTICES">
    <m>
      <v>1 0 0</v>
      ...
    </m>
  </property>
  <property name="FACETS"
            type="SparseMatrix&lt;Rational,NonSymmetric&gt;">
    <m cols="3">
      <v> <e i="1">1</e> </v>
      ...
    </m>
  </property>
  <property name="LINEALITY_SPACE"><m /></property>
  <property name="BOUNDED" value="true" />
  <property name="N_FACETS" value="4" />
  <property name="N_VERTICES" value="4" />
  <property name="VOLUME" value="1/9" />
  <property name="TRIANGULATION">
    <object name="unnamed#0">
      <property name="FACETS">
        <m>
          <v>0 1 2</v>
          <v>1 2 3</v>
        </m>
      </property>
      <property name="F_VECTOR">
        <v>4 5 2</v>
      </property>
    </object>
  </property>
</object>
```

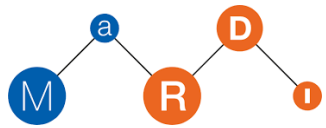
- First version (XML) published in 2016.



The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```

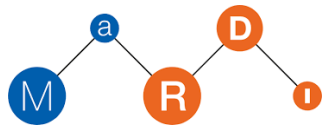

- First version (XML) published in 2016.
- Currently using JSON.



The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```

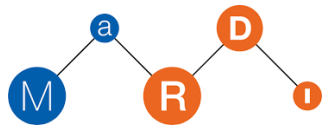

- First version (XML) published in 2016.
- Currently using JSON.
- Builds on already existing infrastructure.



The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```

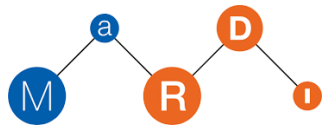

- First version (XML) published in 2016.
- Currently using JSON.
- Builds on already existing infrastructure.
- Extensible, tree structure.



The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```

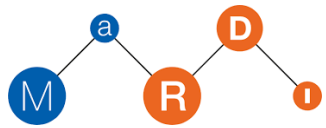

- First version (XML) published in 2016.
- Currently using JSON.
- Builds on already existing infrastructure.
- Extensible, tree structure.
- Has a Schema.



The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```

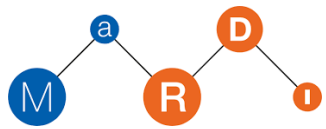

- First version (XML) published in 2016.
- Currently using JSON.
- Builds on already existing infrastructure.
- Extensible, tree structure.
- Has a Schema.
- Older formats will be upgraded on load.



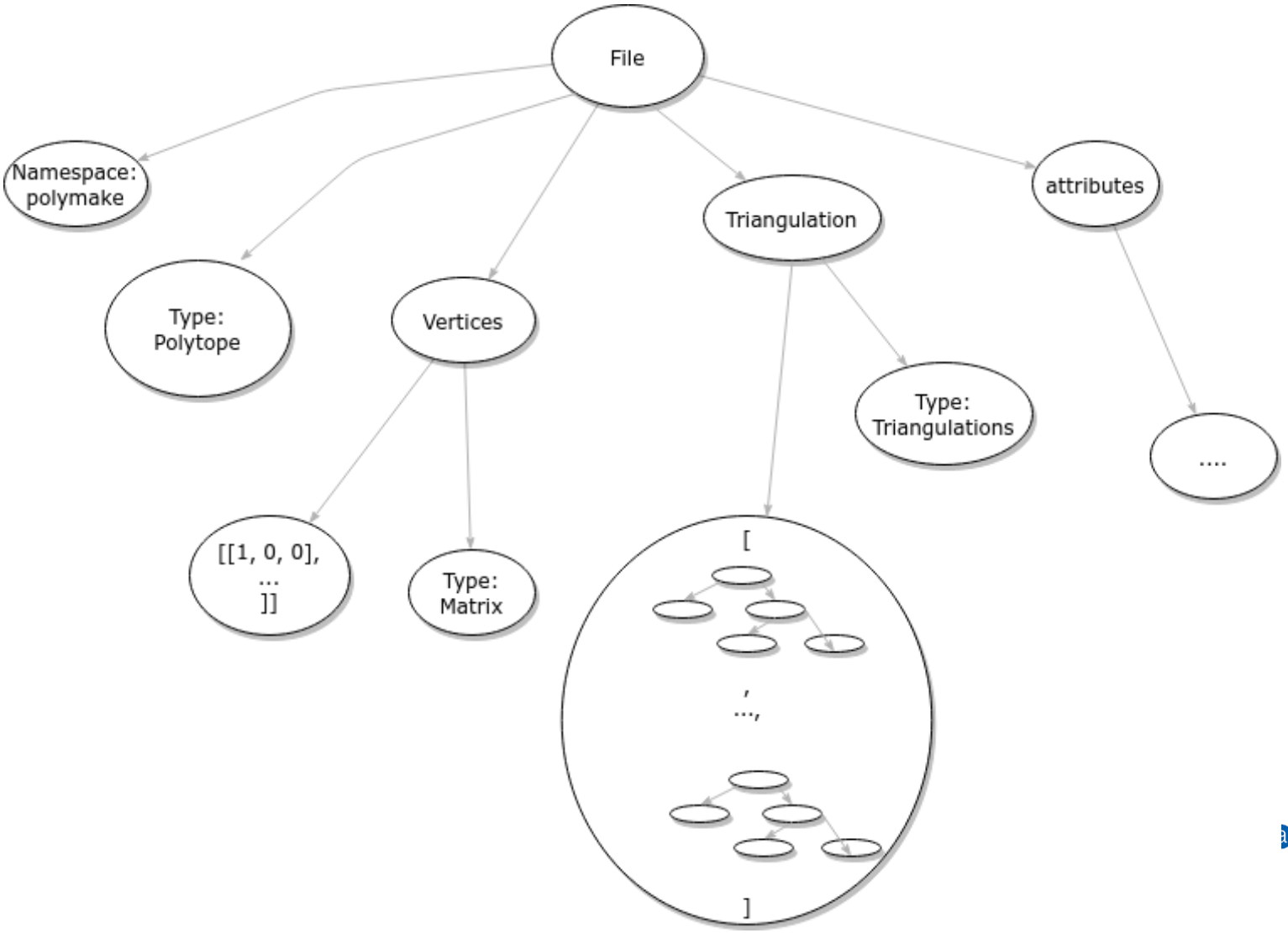
The Polymake File Format


```
{
  "_attrs": {
    "FACETS": {
      "_type": "SparseMatrix<Rational, NonSymmetric>"
    }
  },
  "VERTICES": [["1", "0", "0"],
               ["1", "1/3", "0"],
               ["1", "0", "1/3"],
               ["1", "1/3", "1/3"]],
  "CONE_AMBIENT_DIM": 3,
  "N_VERTICES": 4,
  "_ns": {
    "polymake": [
      "https://polymake.org",
      "4.9"
    ]
  },
  "_info": {
    "description": "cube of dimension 2"
  },
  "TRIANGULATION": [{
      "FACETS": [[ 0, 1, 2],
                 [ 1, 2, 3]],
      "_id": "unnamed#0",
      "F_VECTOR": [4, 5, 2]
  }],
  "N_FACETS": 4,
  "_id": "square",
  "BOUNDED": true,
  "FACETS": [{ "1": "1" }, { "0": "1/3", "1": "-1"},
             { "2": "1" }, { "0": "1/3", "2": "-1" },
             {     "cols": 3    }],
  "_type": "polytope::Polytope<Rational>",
  "VOLUME": "1/9"
}
```


- First version (XML) published in 2016.
- Currently using JSON.
- Builds on already existing infrastructure.
- Extensible, tree structure.
- Has a Schema.
- Older formats will be upgraded on load.
- Can be loaded into OSCAR and SAGE.

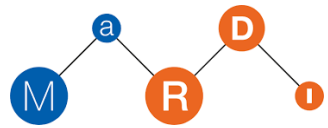


Tree Structure



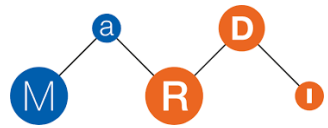
Schemata

- A schema defines a general structure for data so that it can be interpreted by software.



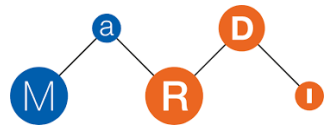
Schemata

- A schema defines a general structure for data so that it can be interpreted by software.
- Schema languages.



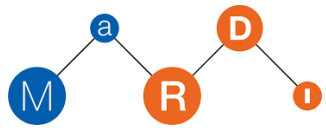
Schemata

- A schema defines a general structure for data so that it can be interpreted by software.
- Schema languages.
- Is possible to define recursive structure.



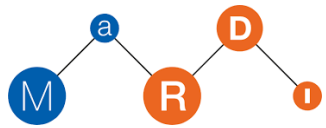
Schemata

- A schema defines a general structure for data so that it can be interpreted by software.
- Schema languages.
- Is possible to define recursive structure.
- Can be generated from software.



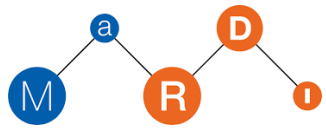
Schemata

- A schema defines a general structure for data so that it can be interpreted by software.
- Schema languages.
- Is possible to define recursive structure.
- Can be generated from software.
- Schemata allow data to be validated without being read into software.



Schemata

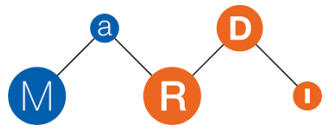
- A schema defines a general structure for data so that it can be interpreted by software.
- Schema languages.
- Is possible to define recursive structure.
- Can be generated from software.
- Schemata allow data to be validated without being read into software.
- Adds structure to document based databases.



On Going Work


```
{
  "_ns": {
    "Oscar": [ ...
    ]
  },
  "type": "QQMPolyRingElem",
  "data": {
    "terms": [
      {
        "coeff": "1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "1", "1", "0" ],
            "entry_type": "Base.Int"
          }
        }
      },
      {
        "coeff": "-1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "0", "0", "2" ],
            "entry_type": "Base.Int"
          }
        }
      }
    ],
    "parent": {
      "type": "QQMPolyRing",
      "data": {
        "base_ring": { "type": "QQField"     },
        "symbols": {
          "type": "Vector",
          "data": {
            "vector": [ "x", "y", "z" ],
            "entry_type": "Symbol"
          }
        }
      }
    }
  }
}
```

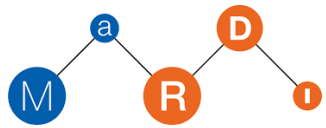

- Prototyping in OSCAR.



On Going Work


```
{
  "_ns": {
    "Oscar": [ ...
    ]
  },
  "type": "QQMPolyRingElem",
  "data": {
    "terms": [
      {
        "coeff": "1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "1", "1", "0" ],
            "entry_type": "Base.Int"
          }
        }
      },
      {
        "coeff": "-1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "0", "0", "2" ],
            "entry_type": "Base.Int"
          }
        }
      }
    ],
    "parent": {
      "type": "QQMPolyRing",
      "data": {
        "base_ring": { "type": "QQField"     },
        "symbols": {
          "type": "Vector",
          "data": {
            "vector": [ "x", "y", "z" ],
            "entry_type": "Symbol"
          }
        }
      }
    }
  }
}
```

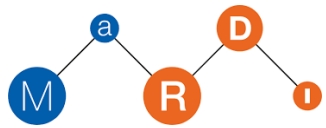

- Prototyping in OSCAR.
- Aim to be software independant.



On Going Work


```
{
  "_ns": {
    "Oscar": [ ...
    ]
  },
  "type": "QQMPolyRingElem",
  "data": {
    "terms": [
      {
        "coeff": "1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "1", "1", "0" ],
            "entry_type": "Base.Int"
          }
        }
      },
      {
        "coeff": "-1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "0", "0", "2" ],
            "entry_type": "Base.Int"
          }
        }
      }
    ],
    "parent": {
      "type": "QQMPolyRing",
      "data": {
        "base_ring": { "type": "QQField"     },
        "symbols": {
          "type": "Vector",
          "data": {
            "vector": [ "x", "y", "z" ],
            "entry_type": "Symbol"
          }
        }
      }
    }
  }
}
```

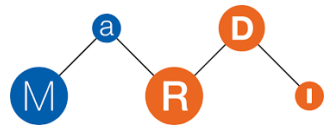

- Prototyping in OSCAR.
- Aim to be software independant.
- Functionality for storing most types in OSCAR.



On Going Work

```
{
  "_ns": {
    "Oscar": [ ...
    ]
  },
  "type": "QQMPolyRingElem",
  "data": {
    "terms": [
      {
        "coeff": "1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "1", "1", "0" ],
            "entry_type": "Base.Int"
          }
        }
      },
      {
        "coeff": "-1",
        "exponent": {
          "type": "Vector",
          "data": {
            "vector": [ "0", "0", "2" ],
            "entry_type": "Base.Int"
          }
        }
      }
    ],
    "parent": {
      "type": "QQMPolyRing",
      "data": {
        "base_ring": { "type": "QQField"     },
        "symbols": {
          "type": "Vector",
          "data": {
            "vector": [ "x", "y", "z" ],
            "entry_type": "Symbol"
          }
        }
      }
    }
  }
}
```

- Prototyping in OSCAR.
- Aim to be software independant.
- Functionality for storing most types in OSCAR.
- Julia package for small databases.



References

Thank You!

Gawrilow, Ewgenij and Hampe, Simon and Joswig, Michael, (2016)

The Polymake XML file format

CoRR

(2002)

RELAX NG Compact syntax specification,

Tech. report, The Organization for the Advancement of Structured Information Standards (OASIS), November 2002,

available at `http://relaxng.org/compact-20021121.html`