# CoCoA-5 - Slug #1597

## GetRow/GetRows are extraordinarily slow

27 May 2021 10:55 - John Abbott

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 27 May 2021 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Anna Maria Bigatti | **% Done:** | 100% |
| **Category:** | enhancing/improving | **Estimated time:** | 3.33 hours |
| **Target version:** | CoCoA-5.4.0 | **Spent time:** | 3.25 hours |

**Description**

For larger matrices GetRow and GetRows are unexpectedly slow:

```
/**/ M := mat([[random(-999,999) | j in 1..1000]]);
/**/ K:=LinKerZZ(M);
/**/ t0 := CpuTime();R := GetRows(K); TimeFrom(t0);
100.424
/**/ t0 := CpuTime();for j := 1 to 1000 do L := GetRow(K,j); EndFor; TimeFrom(t0);
100.046
```

Times are on my Linux box (compiled with debugging active)

**Related issues:**

| | | |
|---|---|---|
| Related to CoCoA-5 - Feature #1296: Matrixrow-functions | **In Progress** | **16 Jun 2019** |
| Related to CoCoA-5 - Slug #31: theValue makes copy | **In Progress** | **15 Nov 2011** |

### History

**#1 - 27 May 2021 11:41 - John Abbott**

*- Status changed from New to In Progress*

*- % Done changed from 0 to 10*

The problem is that **each call** to GetRow makes a copy of the complete matrix!
The copy is triggered by CoCoA::InterpreterNS::theValue.

No wonder it takes ages. But how to avoid the copying?
This might be easy or might be long and tedious... groan!

**#2 - 27 May 2021 11:49 - Anna Maria Bigatti**

This also corresponds to the time for

```
t_0:=CpuTime(); GG:=[[M[i,j] | j in 1..NumCols(M)] | i in 1..NumRows(M)]; TimeFrom(t_0);
```

and also for (discarding RingElem structure)

```
t_0:=CpuTime(); GG:=[[AsINT(M[i,j]) | j in 1..NumCols(M)] | i in 1..NumRows(M)]; TimeFrom(t_0);
```

#### #3 - 27 May 2021 12:06 - John Abbott

*- Related to Feature #1296: Matrixrow-functions added*

#### #4 - 27 May 2021 12:22 - Anna Maria Bigatti

*- Assignee set to Anna Maria Bigatti*

*- Target version changed from CoCoA-5.4.2 to CoCoA-5.4.0*

*- % Done changed from 10 to 90*

Fixed

```
/**/ M := mat([[random(-999,999) | j in 1..1000]]);
/**/ K:=LinKerZZ(M);
/**/ t0 := CpuTime();R := GetRows(K); TimeFrom(t0);
0.377
```

This was/is the problem: the interpreter is obviously slow.
This is the solution for GetRows/Cols: indeed we already had GetRows in CoCoALib, but in CoCoA5 we defined GetRows in mat.cpkg5(!!!!!) using CoCoALib's GetRow.
So I just had to add the CoCoALib to BuiltInOneLiners-CoCoALib.C.
(easy when you know where)

The problem still is: why the interpreter is so slow?

Meanwhile I also fix GetCols.

#### #5 - 27 May 2021 12:54 - Anna Maria Bigatti

Further investigation, this is very surprising for me

```
/**/ M := ZeroMat(ZZ,1000,1000);
/**/ t_0:=CpuTime(); GR:=[5 | i in 1..NumCols(M)]; TimeFrom(t_0);
0.115
/**/ t_0:=CpuTime(); N := NumCols(M); GR:=[5 | i in 1..N]; TimeFrom(t_0);
0.112
/**/ t_0:=CpuTime(); GR:=[5 | i in 1..1000]; TimeFrom(t_0);
0.001
```

So, calling NumCols (of a big matrix) in the interpreter costs a lot.

#### #6 - 28 May 2021 10:47 - Anna Maria Bigatti

```
CoCoA::InterpreterNS::theValue<CoCoA::InterpreterNS::MAT>(boost::intrusive_ptr<CoCoA::InterpreterNS::MAT>)
```

is the function calling

```
CoCoA::matrix::matrix(CoCoA::matrix const&)
```

which makes the copy

#### #7 - 28 May 2021 13:52 - Anna Maria Bigatti

------ **THIS IS NOW COPIED IN** #31 --------

The (template) function theValue makes a copy of

```
typename CoCoALibType<T>::type
```

so I changed it so that it returns

```
const typename CoCoALibType<T>::type&
```

and modified accordingly all its implementations... relying on CVS to recover the original code ;-)

Well, it works, all tests pass.  Too good to be true?

#### #8 - 28 May 2021 13:53 - Anna Maria Bigatti

*- Subject changed from GetRow/GetRows are extraordinarily slow to theValue makes a copy (was: GetRow/GetRows are extraordinarily slow)*

*- Category changed from enhancing/improving to Parser/Interpreter*

*- % Done changed from 90 to 70*

#### #9 - 28 May 2021 15:24 - John Abbott

*- Related to Slug #31: theValue makes copy added*

#### #10 - 28 May 2021 17:58 - Anna Maria Bigatti

*- Subject changed from theValue makes a copy (was: GetRow/GetRows are extraordinarily slow) to GetRow/GetRows are extraordinarily slow*

*- Category changed from Parser/Interpreter to enhancing/improving*

*- Status changed from In Progress to Feedback*

*- % Done changed from 70 to 90*

**#11 - 14 Sep 2021 11:47 - John Abbott**

*- Status changed from Feedback to Closed*

*- % Done changed from 90 to 100*

*- Estimated time set to 3.33 h*