CoCoA-5 - Feature #761

Evaluating a QuasiPol

27 Aug 2015 17:48 - Christof Soeger

Status:	Closed	**Start date:**	27 Aug 2015
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	CoCoA-5 function: new	**Estimated time:**	2.00 hours
Target version:	CoCoA-5.1.3/4 Jan 2016	**Spent time:**	0.85 hour

Description

There should be an CoCoA5 function for evaluating a QuasiPoly. There is on in CoCoALib but we probably cannot use it easily because QuasiPoly is no CoCoA5 type.

History

#1 - 28 Aug 2015 08:58 - Anna Maria Bigatti

- Category set to CoCoA-5 function: new

I don't know how to do it (and if it is possible), but we could return a QuasiPoly in CoCoA-5 as a tagged object (tag "QuasiPoly") this would make it a kind of //CoCoA-5 type//
We'll see next week ;-)

#2 - 02 Sep 2015 18:03 - Christof Soeger

Here is an example and how you have to evaluate it at the moment

```
excl := Mat(ZZ, [
[-1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, 1, 1, -1, 1],
[-1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, 1, -1, -1, 1, 1],
[1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
[1, 1, 1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1],
[1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1]
]);

ineq := Mat(ZZ, [
[-1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, 1, 1, -1, 1],
[-1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, 1, -1, -1, 1, 1]
]);

eq := Mat(ZZ,[[1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1]]);

v := Mat(ZZ,[[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]]);

CondIn := record[excluded_faces := excl, inequalities := ineq, equations := eq, grading := v, signs := v];

T := CpuTime();
C := NmzComputation(CondIn, ["HilbertSeries"]);
"CpuTime = "+DecimalStr(CpuTime()-T) + "s";

Q := C.HilbertQuasiPolynomial;

// Q[0]
eval(Q[1],[0]);

// periode
len(Q);

// Q[13]
eval(Q[2],[13]);
```

I would like to do just eval(Q,13). Since it is no type we probably have to give it a different clear name, like evalQuasiPoly(Q,13).

#3 - 02 Sep 2015 18:17 - Christof Soeger

- % Done changed from 0 to 50

Ups, there is already NmzEvaluateHilbertQuasiPolynomial with exactly that functionality. But it has no documentation.

#4 - 03 Sep 2015 16:38 - Anna Maria Bigatti

added example to CoCoAHelp

#5 - 16 Feb 2016 17:06 - John Abbott

- Status changed from New to Feedback

- % Done changed from 50 to 90

This has been present for over 6 months.

Just one question before closing: why does the fn name contain Hilbert?
While the quasi-poly is typically produced as a Hilbert fn, the fn which evaluates it does not care where the quasi-poly came from.

I would suggest the shorter name NmzEvalQuasiPoly;
we use the abbrev Eval for "evaluate" in a few other cases (*e.g.* EvalHilbertFn);
we use the abbrev Poly for "polynomial" in several cases.

The current name is very long; JAA think it is too long.

#6 - 16 Feb 2016 17:08 - John Abbott

Christof does have a valid point about creating a new "type" (tag?) for QuasiPoly. If we think it is worth considering, we can create a new separate issue (related to this one).

#7 - 16 Feb 2016 17:18 - Christof Soeger

I already suggested EvalQuasiPoly ;)

The QuasiPoly is a general object in CoCoALib and not in the Normaliz interface. So the evaluation function also shouldn't contain Nmz.

#8 - 17 Feb 2016 11:26 - John Abbott

- Status changed from Feedback to Closed

- Assignee set to John Abbott

- % Done changed from 90 to 100

I have renamed the fn to EvalQuasiPoly, and updated the C5 documentation (and the Normaliz tests).
Closing.