CoCoALib - Feature #1488

BuiltIn Interreduce-Function

15 Sep 2020 14:06 - Julian Danner

Status:	Closed	**Start date:**	15 Sep 2020
Priority:	High	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99850	**Spent time:**	3.25 hours

Description

The function interreduce is implemented in CoCoA-5, and the implementation (NotBuiltin.cpkg5) can be translated into C++ in a straightforward way. Below you can find such an implementation which I am already using for more than a year. It is well-tested for lists of polynomials. (As I did not know where to put it best, I just added it to src/CoCoA-5/BuiltInFunctions-CoCoALib.C.)

```
DECLARE_STD_BUILTIN_FUNCTION(ir, 1) { //InterReduce-function added by Danner May'19
  intrusive_ptr<RightValue> w = runtimeEnv->evalArgAs<LIST>(ARG(0));
  vector<RingElem> v = runtimeEnv->evalRVAsListOfRingElem(w, ARG(0));

  //ensure v is not an empty list
  if (v.empty()) {
    return Value::from(v);
  }
  //delete possible zeros in v
  v.erase(std::remove(v.begin(), v.end(), v[0]-v[0]),v.end()); //v[0]-v[0]=0 in the correct ring.

  vector<RingElem> ans;
  RingElem rem;
  int count=0;
  bool newLPPfound=true;
  while(newLPPfound){
    ans=vector<RingElem>();
    if(VerbosityLevel()>=90){printf("interreduced: round n.%i\n", ++count);}
    newLPPfound=false;
    sort(v.begin(), v.end(), [](RingElem elem1, RingElem elem2) {return LPP(elem1)<LPP(elem2);} );
    for(vector<RingElem>::const_iterator it=v.begin(); it!=v.end(); ++it){
      CheckForInterrupt("interreduction");
      rem=NR(*it,ans);
      if(!IsZero(rem)) {
          ans.push_back(rem);
          if(!newLPPfound && LPP(rem)!=LPP(*it)){
              newLPPfound=true;
          }
      }
    }
    v=ans;
  }
  //result is stored in ans
  return Value::from(ans);
}
END_STD_BUILTIN_FUNCTION
```

It would be nice to officially get it in CoCoA-Lib ;)

Related issues:

Related to CoCoALib - Feature #1405: New fn: interreduction	**Closed**	**28 Jan 2020**
Related to CoCoALib - Support #1510: Documentation for SparsePolyOps?	**New**	**14 Oct 2020**
Related to CoCoALib - Design #1642: interreduce: make monic if over finite fi...	**In Progress**	**21 Dec 2021**
Related to CoCoALib - Design #1649: Add file SparsePolyOps-vector.C	**Closed**	**21 Jan 2022**

History

#1 - 16 Sep 2020 14:45 - John Abbott

- Category set to New Function

- Status changed from New to In Progress

- Target version set to CoCoALib-0.99800

- % Done changed from 0 to 10

Here is my revised version of the source code. It could surely be made faster, but this version is relatively simple and "obviously works".

```
  std::vector<RingElem> INTERRED(std::vector<RingElem> v)
  {
    if (v.empty()) { return v; } // ??? or error???
// BUG:  MUST check that all ringelems are in same ring...
    const char* const FnName = "INTERRED";
    VerboseLog VERBOSE(FnName);
    //delete possible zeros in v
    const ring& P = owner(v[0]);
    v.erase(std::remove(v.begin(), v.end(), zero(P)), v.end());

    int count = 0; // BUG?  int or long?
    while (true)
    {
      VERBOSE(90) << "round " << ++count << endl; // NB *always* incrs count!
      const auto CompareLPPs = [](const RingElem& f, const RingElem& g) { return LPP(f)<LPP(g); };
      sort(v.begin(), v.end(), CompareLPPs);

      vector<RingElem> ans;
      RingElem rem;
      bool newLPPfound = false;
      for (const auto& f: v)
      {
        CheckForInterrupt(FnName);
        rem = NR(f,ans);
        if (IsZero(rem)) continue;
        ans.push_back(rem);
        if (!newLPPfound && LPP(rem) != LPP(f))
          newLPPfound = true;
      }
      if (!newLPPfound) return ans;
      swap(v,ans); // quicker than: v = ans;
    }
  }
```

#2 - 16 Sep 2020 15:41 - John Abbott

- % Done changed from 10 to 20

I have now put my version of the impl in a new file **SparsePolyOps-interreduce.C** with corresponding header file.

What name should the function have? And what exactly should its semantics be?
Currently it is called **interreduce** and it computes a new list of polys which is interreduced (the original list is unaffected).

I have not written any doc yet... because we need to answer the 2 questions above first.

NOTE: right the CoCoALib fn is called interreduce but it corresponds to the CoCoA function interreduced :-(

#3 - 16 Sep 2020 16:23 - John Abbott

Might it be useful to sort elements inside the interreduction loop according to a more "sophisticated" ordering: *e.g.*
all monomials are less than all binomials
all binomials are less than all polys with at least 3 terms
within each category (monomial, binomial, longer) sort by LPP

The main operation is NR; is it permitted to rescale the result by a convenient constant?
I tried giving as input a list of random univariates, the result was a single polynomial with LPP=1 (as expected), but the leading coefficient was "nasty".

#4 - 25 Sep 2020 11:20 - John Abbott

- Related to Feature #1405: New fn: interreduction added

#5 - 03 Oct 2020 17:27 - John Abbott

- Status changed from In Progress to Resolved

- Assignee set to John Abbott

- % Done changed from 20 to 70

The function is called **interreduced**: it returns an interreduced **copy** of the original list (which is not changed).

The function interreduce can be implemented as

```
void interreduce(std::vector<RingElem>& v) { swap(v, interreduced(v)); }
```

I do not see any way of doing this better and exception-safely.

QUESTION Should the "smarts" in comment 3 be made into a new issue, or is it simply not that important?

#6 - 14 Oct 2020 21:49 - John Abbott

I have commented out interreduce.
I have renamed the files to SparsePolyOps-interreduced.

NOT YET DOCUMENTED
Where should the doc go? See also #1510

#7 - 29 Jan 2021 11:31 - Anna Maria Bigatti

John Abbott wrote:

> I have commented out interreduce.
> I have renamed the files to SparsePolyOps-interreduced.
>
> **NOT YET DOCUMENTED**
> **Where** should the doc go? See also #1510

This is for me...

#8 - 29 Jan 2021 11:32 - Anna Maria Bigatti

- % Done changed from 70 to 80

#9 - 16 Sep 2021 13:15 - John Abbott

- Related to Support #1510: Documentation for SparsePolyOps? added

#10 - 10 Nov 2021 20:10 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 80 to 90

Is there documentation now?

#11 - 21 Dec 2021 21:56 - John Abbott

- Related to Design #1642: interreduce: make monic if over finite field? added

#12 - 04 Feb 2022 21:37 - John Abbott

This is still not documented -- Anna can you do this?

Should the code be put into SparsePolyOps-vector?
Presumably doc should either be in SparsePolyOps (or SparsePolyOps-vector if that file has separate doc).

#13 - 14 Feb 2022 18:18 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#14 - 14 Feb 2022 18:18 - John Abbott

- Related to Design #1649: Add file SparsePolyOps-vector.C added

#15 - 08 Aug 2022 20:04 - John Abbott

- Priority changed from Normal to High

Anna! Please document and close this issue!

#16 - 30 Nov 2022 18:34 - Anna Maria Bigatti

- Description updated

#17 - 30 Nov 2022 18:36 - Anna Maria Bigatti

Reminder for me: write doc for CoCoALib (and check manual for CoCoA-5)

#18 - 21 Dec 2022 18:04 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

> Reminder for me: write doc for CoCoALib (and check manual for CoCoA-5)

Create the documentation file SparsePolyOps-vector.txt

#19 - 09 Mar 2023 22:25 - John Abbott

Anna?

#20 - 15 Mar 2023 08:40 - Anna Maria Bigatti

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

Anna Maria Bigatti wrote:

> Anna Maria Bigatti wrote:
>
> > Reminder for me: write doc for CoCoALib (and check manual for CoCoA-5)
>
> Create the documentation file SparsePolyOps-vector.txt

done