On reduction in the Boolean term algebra

A talk in RICAM, Linz, March 8, 2006.

S.D.Mechveliani
Program Systems Institute, Pereslavl-Zalessky, Russia,
e-mail: mechvel@botik.ru

1 Acknowledgments

The author is grateful to the supporter of his travel to Linz:

Russian Foundation for Basic Research (grant 06-01-10519),

and supporters of his staying in Linz in March – April 2006
for the Special Semester on Gröbner Bases:

RICAM, Austrian Academy of Sciences, Johannes Kepler University.

2 Example 1: simple task from Group theory

Let `G` be a group and `P` a non-empty subset in `G` such that if `X` and `Y` belong to `P` then `X*(i Y)` belongs to `P`. Prove from this that

(1) `e` $\in$ `P`, **(2)** for all `X` (`X` $\in$ `P ==> i X` $\in$ `P)`

Denote `x` $\in$ `P` as `(P x)`, and create a calculus `groupPCalculus`, with the sort `G`, operators `e, i, *, P`, variables `X, Y, Z`, equations

`(X*Y)*Z = X*(Y*Z), e * X = X, (i X)*X = e`,

equations for the Boolean connectives (we skip their display),
precedence `P >> i >> * >> e >>` Boolean operators. Put `lpo` for the term ordering. The problem is set as

```
groupPCalculus |-
     ( (exist [X] (P X))  &  (forall [X,Y] (P X & P Y ==> P (X*(i Y))))
       ==>
       P e  &  (forall [X] (P X ==> P (i X)))
     )
```

`"&"` is conjunction, `"|"` is disjunction.
Negation, skolemization, and bringing to CNF yields the disjuncts

```
[ P XSk,              not P X | not P Y | P (X*(i Y)),
  not P e | P XSk0,   not P e | not P i XSk0          ],
```

`XSk, XSk0` are the Skolem *constants*. Further, each disjunct converts to BT, with `not x -> x xor 1`, `true -> 1`, `false -> 0`, and so on, the BT simplification rules apply. This yiels the bt-disjuncts

```
[(P XSk) xor 1,             (P (X*(i Y)))&(P Y)&(P X) xor (P Y)&(P X),
 (P XSk0)&(P e) xor (P e),  (P i XSk0)&(P e)
]
```

For a disjunct: `d = 1  <==>  r(d) = 0`, where

`r [] = 1;    r (l | d') = (l+1)&(r d'),     l`   a literal.

```
BTAlgebra = ((Integer/(2))[atoms]) / Ideal[a^2-a | a <- atoms]
```

The refutation is through *unfailing completion*, with extension to BT, and a particular way to use Boolean rules.

Ordering on terms: must be SOTG.

Example: lexicographic path ordering (`lpo`):

```
lpo tab s t =  (s /= t) && lpo' s t                -- how to optimize it?
  where
  lpo' s             (VarT x)        =  varOccurs x [s]
  lpo' (VarT _)      (App _ _ _)     =  False
  lpo' (App f ls rs) (App g ls' rs') =
    let
      lsrs                = ls  ++ rs
      ls'rs'              = ls' ++ rs'
      s_greaterAllSubOf_t = and [lpo tab s s' | s' <- ls'rs']
    in
    or [ or [geqTerm (lpo tab) si t | si <- lsrs]
         ,
         greaterOp tab f g  &&  s_greaterAllSubOf_t
         ,
         f == g  &&  s_greaterAllSubOf_t
         &&
         (case  dropWhile (\ (s, t) -> s == t) $ zip lsrs ls'rs'
          of
          (s, t): _ -> lpo tab s t
          []        -> False
         )
       ]
```

Superposition. Example: derive (1), (2), and do

```
 (1)  (P Y)&(P i Y)&(P e) xor (P Y)&(P e)  |
 (2)  (P i XSk0)&(P e)                     |   -->  (P XSk0)&(P e).
```

by the substitution `[Y := XSk0]`. In the general case, it also applies the complementary factors. This is similar to s-polynomial. The refutation aims to derive `true = false`. There are sufficient only BN-superpositions. The method scheme is of the `RN+` strategy by [Hsi].

3 Comparison to polynomial algebra

`superposition = m1&(sb f) + m2&(sb g)`.
The ground BT algebra is a polynomial ring. `BT<XX>` maps by ground instances.
Example: `(P X) --> P e, (P (i i e)*(i e)), ...`

The main points of difference to polynomial algebra are as follows.

(1) (Internal) variables vs indeterminates.

(2) Partial TRW ordering

(3) Potentially infinite set of indeterminates (a semi-decision procedure).

(4) A substitution may decrease a b-monomial length.
Multiplying by a singular b-monomial may change the order.

(5) The effect of the idempotence law for the indeterminate set.

4 Reduction

[Hsi] suggests reduction only by monomials. For example,

```
Q e & P Y & P (i Y)  xor  P a & R (i e) & P (i a)  xor  P (i Y)  xor  1
```

is reduced by `P X & P (i X)` to `P (i Y) xor 1` by `[X := Y]` and `[X := a]`.

The removal of several monomials is clearly a 'simplification'.

We need to give a reasonable meaning to the notion of
"is simpler b-monomial with respect to equational refutation".

The reduction by monomials is rather weak. For example,

```
P X & P (i X)  xor  P e & P X  xor  P e
```

is 'reduced' by `P Y & P (i X)  xor  P e & P Y` to `P e` by `[Y := X]`,
but this is not reduction by monomial.

Try to provide a stronger BT reduction.
`f --total-ground substitution--> g`. The atoms become totally comparable. Continue to a total ordering on ground monomials.

(R1) `>g>` agrees with reduction by equations.
(R2) It agrees with a non-singular factor (give example).

Reduction method for reducing `h` by a list `fs`.

Convert `h` to ground `g`; reduce `g` by `fs` and convert back.

Reduction of `g` by `fs` is as follows.

`gAtoms =` set of atoms of `g` arranged as one monomial;
`insts = union [relevantInstances gAtoms f | f ∈ fs]`.
Convert `g` and `insts` to `GroundBTerm` (polynomials), obtaining `g0` and `f0s` (note the term ordering extension; apply `degLex` for power products).

Apply Gröbner basis to `f0s ++ [A&A - A | A any atom]` — the version with pre-reduction. In the end, return the result from `GroundBTerm`.

`insts = union [relevantInstances gAtoms f | f ∈ fs]`.

```
relevantInstances gAtoms f = if isGround f then [f]
                             else               [sb f | sb <- gAtomMatches]
```

`coverMon(f) =` minimal set of atoms in `f` that covers `vars f`.
`gAtomMatches =` matches of `coverMon(f)` against subsets of `gAtoms`.

This implements the following ideas.

1. Each `f' = sb f` found above, is ground and related to ground goal.

2. For ground monomials, the graded algebra method works.

3. (1) and (2) make it probable that the Gröbner completion for `Ideal[f' | f' <- insts]` will reduce the goal. If not, still the last basis `f0''` produced in the Gröbner completion often occurs greatly reduced itself, which can be exploited.

4. The whole described reduction (with variables) is a terminating procedure. Though, it is not complete.

5 Questions and problems

(1) Suppose `X` and `Y` convert to the indeterminates `Xc >> Yc`, and the Gröbner method has reduced `P Xc & P e` to `P Yc & P e`.

Here `P Xc >> P Yc >> P e` are the three ground atoms — indeterminates. Then, the variables return, and the final reduction is `P Y & P e`. Is the latter BT 'simpler' than the initial one?

(2) Matches of `coverMon(f)` to subsets of `gAtoms`. may cost `O( factorial(|gAtoms|) )`.

(3) The Gröbner completion costs about $O(d^{n^2})$.

(4) The order in which new constants in `g` include into the symbol precedence may bring a great difference.

We also apply the *resource* bound.

6 BT reduction based on multiset ordering

```
mon >mls> mon' =  gtm (mon \\ mon') (mon' \\ mon)
                               -- cancel the set intersection
 where
 gtm _  [] = True
 gtm [] _  = False
 gtm m  n  = exist [a <- m, b <- n]  such that
                                a >> b  and  m >mls> (n \\ [b])
```

`>mls>` agrees with the TRW reduction on all monomials (not only on ground):

(Mls1) `>mls>` agrees with the atom reduction.

(Mls2) If `m >mls> m'`, and `a` is not in `m`, then `(a & m) >mls> (a & m')`.

(Mls3) `>mls>` is invariant under regular substitutions.

(Mls4) `>mls>` is terminating (Noetherian).

Naturally, `>mls>` continues to the multiset ordering on BT.

`matchDivide m' m =  Just (sb, q)` or `Nothing`.

Reduction method for a BT `f` by a BT list `gs`

Find reductive `g` in `gs` and reduce `f` with it, and continue.
Reducing a BT `f` by a BT `g` is as follows.
`maxs =` maximal by `>mls>` monomials in `g`;
`m` $\in$ `maxs`.
Find `m'` in `f` such that `mathchDivide m' m = Just (sb, q)`, (`m' = q*(sb m)`), and then put
`f  ->  f' = f - q*(sb g)`, if this is reductive in `>mls>`.

Corollary.
If `m` is greatest in `g`, and `m` match-divides `m'` in `f` to `(sb, q)`, with a regular `sb` for `m`, then `f - q*(sb g)` is a reduction.

Example 1: a monomial `g`.

Example 2: case of `m` being a multiple of other monomials in `g`.

7 Practice with the described reductions

On March 2006, we apply only the simplest BT reduction
"if `f` is in `gs` then 0 else reduce by monomials".

Next, we are going to consider the reduction by the multiset ordering.

Then, we could also think of how to make more realistic the method of Groebner basis for the ground relevant instances.

References

[B:D:P] L. Bachmair, N. Dershovitz, D. Plaisted.: *Completion without failure.* In Ait-Kaci (Editor): Resolution of Equations in Algebraic Structures, Volume 2, pages 1-30, Academic Press, 1989.

[Bu] B. Buchberger. *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory.* CAMP. Publ. No.83–29.0 November 1983.

[Hsi] J. Hsiang. *Refutational theorem proving using term-rewriting systems.* Artificial Intelligence, 1985, v.25, p.255-300.

[Hsi:Ru] J. Hsiang, M. Rusinowitch. *On word problems in equational theories.* In Th. Ottman (ed.), Proceedings of the Fourteenth International Conference on Automata, Languages and Programming, Karlsruhe, West Germany, July 1987, Springer Verlag, Lecture Notes in Computer Science **267**, (54–71), 1987.