CERTIFYING IDEAL MEMBERSHIP TESTS

Daniela Kaufmann

TU Wien, Austria

Dagstuhl Seminar 25231 "Certifying Algorithms for Automated Reasoning"
Schloss Dagstuhl, Wadern, Germany

June 2, 2025

TU WIEN Informatics

FWF Austrian Science Fund

Ideal Membership Problem

Given $x - y$ and $x - 2 \in \mathbb{Q}[x, y]$.

Is $f = xy + x - 3y$ a consequence of these two equations?

Ideal Membership Problem

Given $x - y$ and $x - 2 \in \mathbb{Q}[x, y]$.

Is $f = xy + x - 3y$ a consequence of these two equations?

$x - y = 0 \wedge x - 2 = 0 \implies xy + x - 3y = 0$?

Ideal Membership Problem

Given $x - y$ and $x - 2 \in \mathbb{Q}[x, y]$.

Is $f = xy + x - 3y$ a consequence of these two equations?

$x - y = 0 \wedge x - 2 = 0 \implies xy + x - 3y = 0$?

Algebraically: $xy + x - 3y \in \langle x - y, x - 2 \rangle$?

Ideal

Ideal. A subset $I \subset R[X]$ is an ideal if it satisfies:

- $0 \in I$
- If $f, g \in I$, then $f + g \in I$.
- If $f \in I$ and $h \in R[X]$ then $hf \in I$.

Ideal

Ideal. A subset $I \subset R[X]$ is an ideal if it satisfies:

- $0 \in I$
- If $f, g \in I$, then $f + g \in I$.
- If $f \in I$ and $h \in R[X]$ then $hf \in I$.

Basis.
Let $f_1, \ldots, f_s \in R[X]$. Then we set

$$\langle f_1, \ldots, f_s \rangle = \{h_1 f_1 + \cdots + h_s f_s \mid h_1, \ldots, h_s \in R[X]\}.$$

$\langle f_1, \ldots, f_s \rangle$ is an ideal and is called the ideal generated by $f_1, \ldots, f_s$.

Hilbert Basis Theorem. Every ideal has a finite basis.

Interpretation

The ideal $\langle f_1, \ldots, f_s \rangle$ has a nice interpretation in terms of polynomial equations.

Given $f_1, \ldots, f_s \in R[X]$, we get the system of equations

$$\begin{aligned} f_1 &= 0, \\ &\vdots \\ f_s &= 0. \end{aligned}$$

Interpretation

The ideal $\langle f_1, \ldots, f_s \rangle$ has a nice interpretation in terms of polynomial equations.

Given $f_1, \ldots, f_s \in R[X]$, we get the system of equations

$$\begin{aligned} f_1 &= 0, \\ &\vdots \\ f_s &= 0. \end{aligned}$$

Let $h_1, \ldots, h_s \in R[X]$. We can derive $h_1 f_1 = 0$, $h_2 f_2 = 0$, $h_1 f_1 + h_2 f_2 = 0$ etc.

Hence we obtain $h_1 f_1 + \cdots + h_s f_s = 0$ as a consequence of our initial system.

Interpretation

The ideal $\langle f_1, \ldots, f_s \rangle$ has a nice interpretation in terms of polynomial equations.

Given $f_1, \ldots, f_s \in R[X]$, we get the system of equations

$$f_1 = 0,$$
$$\vdots$$
$$f_s = 0.$$

Let $h_1, \ldots, h_s \in R[X]$. We can derive $h_1 f_1 = 0$, $h_2 f_2 = 0$, $h_1 f_1 + h_2 f_2 = 0$ etc.

Hence we obtain $h_1 f_1 + \cdots + h_s f_s = 0$ as a consequence of our initial system.

Thus, we can think of $\langle f_1, \ldots, f_s \rangle$ as consisting of all "polynomial consequences" of the equations $f_1 = f_2 = \ldots = f_s = 0$.

Why do we care about ideal membership?

- **Solving Polynomial Systems:** Check if a polynomial follows from a system of equations
- **Automated Theorem Proving:** Prove algebraic properties and relationships from given axioms
- **Formal Verification:** Check whether specification is implied from a model defined by polynomial equations
- **Algebraic Geometry:** Determine whether a polynomial vanishes on a variety defined by an ideal
- **Computer Algebra:** Simplify expressions and perform reductions using ideal membership
- ...

Ideal Membership Problem

Let $I = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Ideal Membership Problem

Let $I = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Spoiler: Yes, because

$$xy + x - 3y = (x - y) + y(x - 2)$$

Ideal Membership Problem

Let $I = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

We need some kind of division/reduction algorithm $f \xrightarrow{\{f_1,\dots f_s\}} r$ to check this.

Leading Elements

Let f in $R[x_1, \ldots, x_n]$ be ordered w.r.t to an ordering $<$ such that

$$f = a_1\tau_1 + a_2\tau_2 + \ldots + a_m\tau_m.$$

Then we call

- $\mathrm{lt}(f) = a_1\tau_1$ is the **leading term** of f.
- $\mathrm{lm}(f) = \tau_1$ is the **leading monomial** of f.
- $\mathrm{lc}(f) = a_1$ is the **leading coefficient** of f.
- $f - \mathrm{lt}(f) = a_2\tau_2 + \ldots + a_m\tau_m$ is the **tail** of f.

Algorithm $f \xrightarrow{\{f_1,\dots f_s\}} r$

Input: $f_1, \dots, f_s, f$
Output: $a_1, \dots, a_s, r$
$a_1 := 0; \dots; a_s := 0; r := 0$
$p := f$
WHILE $p \neq 0$ DO
    $i := 1$
    divisionoccurred := false
    WHILE $i \leq s$ AND divisionoccurred = false DO
        IF $\mathrm{LT}(f_i)$ divides (p) THEN
            $a_i := a_i + \mathrm{LT}(p)/\mathrm{LT}(f_i)$
            $p := p - (\mathrm{LT}(p)/\mathrm{LT}(f_i))f_i$
            divisionoccurred:= true
        ELSE
            $i := i + 1$
    IF divisionoccurred = false THEN
        $r := r + \mathrm{LT}(p)$
        $p := p - \mathrm{LT}(p)$

*Cox, Little, O'Shea: Ideals, Varieties, and Algorithms, 4th edition, Springer 2015

Ideal Membership Problem

Let $I = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Ideal Membership Problem

Let $I = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

$$xy + x - 3y \xrightarrow{x-y} y^2 - 2y$$

$$xy + x - 3y \xrightarrow{x-2} y - 2$$

Gröbner Basis

Because of Gröbner bases the ideal membership problem is decidable:

- Every ideal $I \subseteq R[X]$ has a Gröbner basis G w.r.t. a fixed monomial order.
- Buchberger's algorithm computes a Gröbner basis $G = \{g_1, \dots, g_m\}$ for the ideal $\langle f_1, \dots, f_s \rangle$.
- Given a Gröbner basis G, there is a computable function $\mathrm{red}_G \colon R[X] \to R[X]$ such that $\forall\, f \in R[X] : \mathrm{red}_G(f) = 0 \iff f \in \langle G \rangle$.
- If $f, r \in R[X]$ are such that $\mathrm{red}_G(f) = r$, then there exist $h_1, \dots, h_m \in R[X]$ such that $f - r = h_1 g_1 + \cdots + h_m g_m$.

Gröbner Bases

Gröbner basis. Fix a monomial order. A finite subset $G = \{g_1, \ldots, g_t\} \subset R[x_1, \ldots, x_n]$ of an ideal $I \subset R[x_1, \ldots, x_n]$ is said to be a Gröbner basis if $\langle \mathrm{lt}(g_1), \ldots, \mathrm{lt}(g_t) \rangle = \langle \mathrm{lt}(I) \rangle$.

Buchberger's Criterion. G is a Gröbner basis of the ideal I if and only if the remainder of the division of $\mathrm{spol}(p, q)$ by G is zero for all pairs $(p, q) \in G \times G$.

S-Polynomials. We define S-polynomials

$$\mathrm{spol}(p, q) \;:=\; \frac{\mathrm{lcm}(\mathrm{lt}(p), \mathrm{lt}(q))}{\mathrm{lm}(p)} p - \frac{\mathrm{lcm}(\mathrm{lt}(p), \mathrm{lt}(q))}{\mathrm{lm}(q)} q$$

for all $p, q \in R[x_1, \ldots, x_n] \setminus \{0\}$, with lcm the least common multiple.

Computing a Gröbner Basis

Algorithm: Buchberger's Algorithm

Input : $F = \{f_1, \ldots, f_s\}$, monomial ordering $<$
Output: Gröbner basis $G = \{g_1, \ldots, g_t\}$ w.r.t. $<$, such that $\langle f_1, \ldots, f_s\rangle = \langle g_1, \ldots, g_t\rangle$

$G = F$;
$C = \{\{g_1, g_2\} \mid g_1, g_2 \in G, g_1 \neq g_2\}$;
while not all pairs $\{g_1, g_2\} \in C$ are marked **do**
    choose unmarked pair $\{g_1, g_2\}$;
    mark $\{g_1, g_2\}$;
    $h =$ normalform of $\mathrm{spol}(g_1, g_2)$ w.r.t. G ($\mathrm{spol}(g_1, g_2) \xrightarrow{G} h$);
    if $h \neq 0$ **then**
        $C = C \cup \{\{g, h\} \mid g \in G\}$;
        $G = G \cup \{h\}$;
return G

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Calculate a Gröbner basis G of I:

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Calculate a Gröbner basis G of I:
$\mathrm{spol}(f_1, f_2) = f_1 - f_2 = y - 2 =: f_3$, so $G = \{f_1, f_2, f_3\}$.

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Calculate a Gröbner basis G of I:
$\mathrm{spol}(f_1, f_2) = f_1 - f_2 = y - 2 =: f_3$, so $G = \{f_1, f_2, f_3\}$.
$\mathrm{spol}(f_1, f_3) = yf_1 - xf_3 = 2x - y^2 \xrightarrow{f_2} y^2 - 4 \xrightarrow{f_3} 0$
$\mathrm{spol}(f_2, f_3) = yf_2 - xf_3 = 2x - 2y \xrightarrow{f_1} 0$

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Calculate a Gröbner basis G of I:
$\text{spol}(f_1, f_2) = f_1 - f_2 = y - 2 =: f_3$, so $G = \{f_1, f_2, f_3\}$.
$\text{spol}(f_1, f_3) = yf_1 - xf_3 = 2x - y^2 \xrightarrow{f_2} y^2 - 4 \xrightarrow{f_3} 0$
$\text{spol}(f_2, f_3) = yf_2 - xf_3 = 2x - 2y \xrightarrow{f_1} 0$

Gröbner(f_1, f_2) = $G = \{x - y, x - 2, y - 2\}$

Ideal Membership Problem

Let $I = \langle f_1, f_2 \rangle = \langle x - y, x - 2 \rangle \subset \mathbb{Q}[x, y]$.
Is the polynomial $f = xy + x - 3y \in I$?

Calculate a Gröbner basis G of I:
$\mathrm{spol}(f_1, f_2) = f_1 - f_2 = y - 2 =: f_3$, so $G = \{f_1, f_2, f_3\}$.
$\mathrm{spol}(f_1, f_3) = yf_1 - xf_3 = 2x - y^2 \xrightarrow{f_2} y^2 - 4 \xrightarrow{f_3} 0$
$\mathrm{spol}(f_2, f_3) = yf_2 - xf_3 = 2x - 2y \xrightarrow{f_1} 0$

Gröbner(f_1, f_2) = $G = \{x - y, x - 2, y - 2\}$

$$xy + x - 3y \xrightarrow{x-y} y^2 - 2y \xrightarrow{y-2} 0$$

$$xy + x - 3y \xrightarrow{x-2} y - 2 \xrightarrow{y-2} 0$$

How to derive a certificate for $\mathrm{spol}(f_i, f_j)$ **and** $f \xrightarrow{\{f_1,\ldots,f_s\}} r$**?**

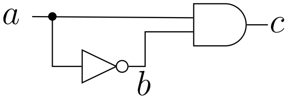
Nullstellensatz Proofs

```
Input: f_1, ..., f_s, f
Output: a_1, ..., a_s, r
a_1 := 0; ...; a_s := 0; r := 0
p := f
WHILE p ≠ 0 DO
        i := 1
        divisionoccurred := false
        WHILE i ≤ s AND divisionoccurred = false DO
                IF LT(f_i) divides (p) THEN
                        a_i := a_i + LT(p)/LT(f_i)
                        p := p − (LT(p)/LT(f_i)) f_i
                        divisionoccurred:= true
                ELSE
                        i := i + 1
        IF divisionoccurred = false THEN
                r := r + LT(p)
                p := p − LT(p)
```

- Provides list of co-factors $a_1, \ldots, a_s$.
- Correctness is checked by expanding linear combination $f = \sum a_i f_i$.
- Condensed proof format.
- Not ideal for debugging.

Beame, P., Impagliazzo, R., Krajicek, J., Pitassi, T., Pudlák, P.: Lower Bounds on Hilbert's Nullstellensatz and Propositional Proofs. In: Proc. London Math. Society. vol. s3-73, pp. 1-26 (1996)

Example



$$
\begin{aligned}
G &= \{ & -b + 1 - a, \quad & b = \neg a \\
&& -c + ab, \quad & c = a \wedge b = a \wedge \neg a \\
&& a^2 - a, \} \quad & a \in \mathbb{B} \\
f &= & c & \\
\\
\mathrm{red}_G(f) &= & 0 &
\end{aligned}
$$

$$f = c = a(-b + 1 - a) - 1(-c + ab) + 1(a^2 - a)$$

$$P = (a, -1, 1)$$

Polynomial Calculus*

Let $G \subseteq R[X]$ and $f \in R[X]$.

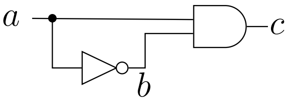
Proof: Sequence $P = (p_1, \ldots p_n)$, where each p_k is obtained by one of the two rules:

Addition	$\dfrac{p_i \qquad p_j}{p_i + p_j}$	p_i, p_j appearing earlier in the proof or are contained in G
Multiplication	$\dfrac{p_i}{qp_i}$	p_i appearing earlier in the proof or is contained in G and $q \in R[X]$ being arbitrary

If $p_n = f$ we have $f \in \langle G \rangle$.

Clegg, M., Edmonds, J., Impagliazzo, R.: Using the Groebner basis algorithm to find proofs of unsatisfiability. In: STOC. pp. 174–183. ACM (1996)

Example



$$\begin{aligned} G &= \{ && -b+1-a, && b = \neg a \\ & && -c+ab, && c = a \wedge b = a \wedge \neg a \\ & && a^2-a, \} && a \in \mathbb{B} \\ f &= && c \end{aligned}$$

$$\mathrm{red}_G(f) = 0$$

$$\begin{array}{llll} * & -b+1-a & & \\ \hline + & -ab+a-a^2 & a^2-a & \\ \hline & + & -ab & -c+ab \\ \hline & & * & -c \\ \hline & & & c \end{array}$$

$$P = (-ab+a-a^2,\ -ab,\ -c,\ c)$$

Practical Algebraic Calculus

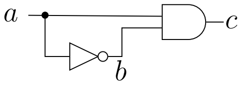
We translate the polynomial calculus into a more concrete proof format:

- For correctness it is important to know how the polynomials in the proof where derived
- Usually known $\rightarrow$ store this information

Practical Algebraic Calculus (PAC)

allows automated proof checking

Practical Algebraic Calculus - SC^2 2018



P =	-b+1-a,	$b = \neg a$
	-c+a*b,	$c = a \wedge b = a \wedge \neg a$
	-a^2+a	$a = \perp \vee a = \top$
Target =	c	$c = \perp$

```
* : -b+1-a,       a,        -a*b+a-a^2;
* : -a^2+a,       -1,       a^2-a;
+ : -a*b+a-a^2,   a^2-a,    -a*b;
+ : -a*b,         -c+a*b,   -c;
* : -c,           -1,       c;
```

D. Ritirc, A. Biere, M. Kauers, A Practical Polynomial Calculus for Arithmetic Circuit Verification, SC2-Workshop, 2018.

Proof Checking

A proof **rule** contains four components:

$$o : v, w,\ p;$$

Proof checking:

- Connection property: v, w are given polynomials or conclusions p_i of previous rules
- Inference property: verify correctness of each rule, e.g. $p = v + w$ for $o =$ "$+$"

- Target check: at least one p_i is equal to f.

Proof Checking Algorithm

input G sequence of given polynomials
$r_1 \cdots r_k$ sequence of PAC proof rules

output "incorrect", "correct-proof", or "correct-refutation"

$P_0 \leftarrow G$
for $i \leftarrow 1 \ldots k$
  let r_i = (o_i, v_i, w_i, p_i)
  case o_i = $+$
    if $v_i \in P_{i-1} \;\wedge\; w_i \in P_{i-1} \;\wedge\; p_i = v_i + w_i$ **then** $P_i \leftarrow$ append(P_{i-1}, p_i)
    else return "incorrect"
  case o_i = $*$
    if $v_i \in P_{i-1} \;\wedge\; p_i = v_i * w_i$ **then** $P_i \leftarrow$ append(P_{i-1}, p_i)
    else return "incorrect"
for $i \leftarrow 1 \ldots k$
  if $p_i = f$ **then return** "target-checked"
return "correct-proof"

Engineering

Use `PolynomialReduce` for reduction $f \xrightarrow{G} 0$ and deriving co-factors $h_1, ..., h_k$ such that $f = h_1 g_1 + \ldots + h_k g_k$.

First we generate a multiplication proof rule for each product $h_i g_i$.

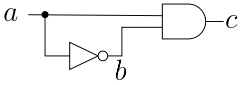
$$* : g_1, h_1, h_1 g_1; \qquad \cdots \qquad * : g_k, h_k, h_k g_k;$$

These products are now simply added together as follows:

$$\begin{array}{llll} + : & h_1 g_1, & h_2 g_2, & h_1 g_1 + h_2 g_2; \\ + : & h_1 g_1 + h_2 g_2, & h_3 g_3, & h_1 g_1 + h_2 g_2 + h_3 g_3; \\ & & \vdots & \\ + : & h_1 g_1 + \ldots + h_{k-1} g_{k-1}, & h_k g_k, & f; \end{array}$$

Boolean Variables - FMCAD 2020

Handle Boolean-value constraints implicitly to reduce number of proof steps.




```
   P =   -b+1-a,
         -c+a*b,
         -a^2+a
Spec =   c
```

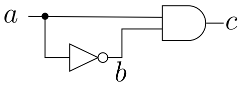
~~-a^2+a~~

```
* : -b+1-a,      a,        -a*b;
+ : -a*b,        -c+a*b,   -c;
* : -c,          -1,       c;
```

D. Kaufmann, M. Fleury, A. Biere, The Proof Checkers Pacheck and Pastèque for the Practical Algebraic Calculus, FMCAD, 2020.

Indices - FMCAD 2020

Introduce indices to reduce proof size.



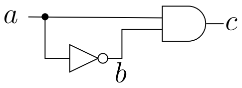

```
P =    1 -b+1-a;
       2 -c+a*b;
Spec = c
```

```
3   * 1,   a,   -a*b;
4   + 3,   2,   -c;
5   * 4,  -1,   c;
```

D. Kaufmann, M. Fleury, A. Biere, The Proof Checkers Pacheck and Pastèque for the Practical Algebraic Calculus, FMCAD, 2020.

Deletion Rule - FMCAD 2020

Introduce a deletion rule to reduce the memory usage of the proof checker.




```
   P =   1 -b+1-a;
         2 -c+a*b;
Spec =   c
```

```
3   * 1,   a,   -a*b;
1   d;
4   + 3,   2,   -c;
2   d;
3   d;
5   * 4,  -1,   c;
```

D. Kaufmann, M. Fleury, A. Biere, The Proof Checkers Pacheck and Pastèque for the Practical Algebraic Calculus, FMCAD, 2020.

Extension Rule - FMCAD 2020

The extension rule allows to add model preserving polynomials to the constraint set.

$$\frac{\overline{x} \vee \overline{y} \qquad y \vee z}{\overline{x} \vee z}$$

Extension Rule - FMCAD 2020

The extension rule allows to add model preserving polynomials to the constraint set.

$$\frac{xy \qquad (1-y)(1-z)}{x(1-z)}$$

D. Kaufmann, M. Fleury, A. Biere, *The Proof Checkers Pacheck and Pastèque for the Practical Algebraic Calculus*, FMCAD, 2020.

Extension Rule - FMCAD 2020

The extension rule allows to add model preserving polynomials to the constraint set.

$$\frac{xy \qquad (1-y)(1-z)}{x(1-z)}$$

```
   P =  1 x*y;
        2 y*z-y-z+1;
Spec =  -x*z+x
```

```
3  =  f, -z+1;
4  *  3,  y-1, -f*y+f-y*z+y+z-1;
5  +  2,    4, -f*y+f;
```

Ext(i, v, p) $\quad (X, P) \Rightarrow (X \cup \{v\}, P(i \mapsto -v + p))$
provided that $P(i) = \perp$ and $v \notin X$ and $p \in \mathbb{Z}[X]/\langle B(X)\rangle$,
and $p^2 - p \equiv 0 \mod \langle B(X)\rangle$.

Extension Rule - FMCAD 2020

The extension rule allows to add model preserving polynomials to the constraint set.

$$\frac{xy \qquad (1-y)(1-z)}{x(1-z)}$$

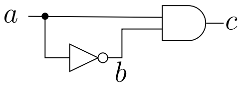
```
   P =   1 x*y;
         2 y*z-y-z+1;
Spec =   -x*z+x
```

```
 3  =  f, -z+1;
 4  *  3,  y-1, -f*y+f-y*z+y+z-1;
 5  +  2,    4, -f*y+f;
 6  *  1,    f, f*x*y;
 7  *  5,    x, -f*x*y+f*x;
 8  +  6,    7, f*x;
 9  *  3,    x, -f*x-x*z+x;
10  +  8,    9, -x*z+x;
```

D. Kaufmann, M. Fleury, A. Biere, The Proof Checkers Pacheck and Pastèque for the Practical Algebraic Calculus, FMCAD, 2020.

LPAC - FMSD 2022

Switch from explicit addition and multiplication rules to linear combinations.



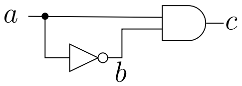

```
P    =  1 -b+1-a;
        2 -c+a*b;
Spec =  c
```

LPAC simulates PAC

```
3  % 1*(a), -a*b;
1  d;
4  % 3*(1)+2*(1), -c;
2  d;
3  d;
5  % 4*(-1), c;
```

LPAC - FMSD 2022

Switch from explicit addition and multiplication rules to linear combinations.




```
   P =  1 -b+1-a;
        2 -c+a*b;
Spec =  c
```

LPAC simulates PAC

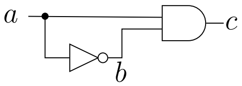
```
3   % 1*(a), -a*b;
1   d;
4   % 3*(1)+2*(1), -c;
2   d;
3   d;
5   % 4*(-1), c;
```

LPAC

```
3   % 1*(a)+2*(1), -c;
1   d;
2   d;
4   % 4*(-1), c;
```

LPAC - FMSD 2022

Switch from explicit addition and multiplication rules to linear combinations.




```
   P =   1 -b+1-a;
         2 -c+a*b;
Spec =   c
```

LPAC simulates PAC

```
3    % 1*(a), -a*b;
1    d;
4    % 3*(1)+2*(1), -c;
2    d;
3    d;
5    % 4*(-1), c;
```

LPAC

```
3    % 1*(a)+2*(1), -c;
1    d;
2    d;
4    % 4*(-1), c;
```

LPAC simulates NSS

```
3    % 1*(-a)+2*(-1), -c;
```

D. Kaufmann, M. Fleury, A. Biere, M. Kauers, Practical Algebraic Calculus and Nullstellensatz with the Checkers Pacheck and Pastèque and Nuss-Checker FMSD, 2022.

PACHECK2

- supports new and old PAC format

- PACHECK2 reads three input files `<input>`, `<proof>`, and `<target>`.
- Verifies that the polynomial in `<target>` is contained in the ideal generated by the polynomials in `<input>` using the rules provided in `<proof>`.

PASTÈQUE2

written by M. Fleury

Theorem Prover Isabelle/HOL

Refinement Approach, relying on Isabelle's Refinement Framework

- abstract specification on ideals: specification in ideal
- final step: executable checker

Isabelle's Archive of Formal Proofs 8000 lines of code

Evaluation: Circuit Verification

PACHECK2

		LPAC simulates PAC			LPAC			LPAC simulates NSS		
multiplier	n	steps	sec	MB	steps	sec	MB	steps	sec	MB
btor	128	0.3	5	94	0.1	2	94	1	2	98
btor	256	1.3	26	367	0.3	8	367	1	7	385
btor	512	5.2	149	1468	1.0	37	1496	1	35	1555

PASTÈQUE2

		LPAC simulates PAC			LPAC			LPAC simulates NSS		
multiplier	n	steps	sec	MB	steps	sec	MB	steps	sec	MB
btor	128	0.3	14	1305	0.1	7	1305	1	53	2044
btor	256	1.3	67	3467	0.3	37	3816	1	762	8819
btor	512	5.2	351	14651	1.0	238	16173	1	14347	41712

LPAC with Proof Recycling – ongoing work

Let $G = \{x - 2y + 2,\ y - z - 1,\ a - b,\ b + z,\ a + b + x + 1\}$. We certify $1 \in \langle G \rangle$:

```
g1  x-2y+2;
g2  y-z-1;
g3  a-b;
g4  b+z;
g5  a+b+x+1;

PNew (1, [ (p1 v1-2*v2), (p2 v2-v3) ], [ (p3 % p1 + p2*(2),  v1 - 2v3) ], {p3})

PApply (1, [v1 = x, v2 = y-1, v3 = z], (g1, g2), {(g6, x - 2z)})
PApply (1, [v1 = a+b, v2 = b, v3 = -z],(g3, g4), {(g7, a + b + 2z)})

g8  % g5 + g6*(-1) + g7*(-1), 1;
```

D. Kaufmann and C. Hofstadler, Recycling Algebraic Proof Certificates, Submitted to SC2-Workshop, 2025.

Shorter Proofs

Clemens Hofstadler and Thibaut Verron,
Short proofs of ideal membership,
Journal of Symbolic Computation, Volume 125, 2024

- Their application: Short proofs of ideal membership for noncommutative settings.
- All results also apply to the commutative case!

A not so trivial example

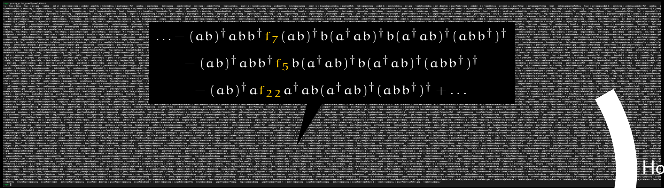
Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

Proof

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$



$$\ldots - (ab)^\dagger abb^\dagger f_7 (ab)^\dagger b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$
$$- (ab)^\dagger abb^\dagger f_5 b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$
$$- (ab)^\dagger a f_{22} a^\dagger ab(a^\dagger ab)^\dagger (abb^\dagger)^\dagger + \ldots$$

How?

Another proof

$$\begin{aligned}(ab)^\dagger - b^\dagger a^\dagger = {} & f_{21} - f_{10} + b^\dagger f_{14} - f_{12}(ab)^\dagger - b^\dagger(abb^\dagger)^\dagger f_{11} + b^\dagger(abb^\dagger)^\dagger f_{15} \\ & + (a^\dagger ab)^\dagger a^\dagger f_9 (ab)^\dagger - b^* f_{23}((ab)^\dagger)^*(ab)^\dagger - f_{21} ab(ab)^\dagger + f_{22} ab(ab)^\dagger \\ & - f_{39}(a^\dagger)^*((ab)^\dagger)^*(ab)^\dagger + b^\dagger(abb^\dagger)^\dagger((abb^\dagger)^\dagger)^*(b^\dagger)^* f_{31} - b^\dagger f_{14} d^* b^*(a^\dagger)^* \\ & + (a^\dagger ab)^\dagger a^\dagger ab f_{12}(ab)^\dagger - b^\dagger(abb^\dagger)^\dagger f_{15}((ab)^\dagger)^* b^*(a^\dagger)^* \\ & + f_{20} b^*(a^\dagger)^*((ab)^\dagger)^*(ab)^\dagger + (a^\dagger ab)^\dagger a^\dagger abb^* f_{23}((ab)^\dagger)^*(ab)^\dagger\end{aligned}$$

2

Shorter Proofs

Problem

Given $f, f_1, \ldots, f_s \in R[X]$, $N \in \mathbb{N}$

Compute $a_i \in \langle X \rangle$, $c_i \in R$ such that

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

if existent, else return FAILED.

Shorter Proofs

Problem

Given $f, f_1, \ldots, f_s \in R[X]$, $N \in \mathbb{N}$

Compute $a_i \in \langle X \rangle$, $c_i \in R$ such that

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

if existent, else return FAILED.

Question Is this problem decidable?

Shorter Proofs

Problem

Given $f, f_1, \ldots, f_s \in R[X]$, $N \in \mathbb{N}$

Compute $a_i \in \langle X \rangle$, $c_i \in R$ such that

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

if existent, else return FAILED.

Question Is this problem decidable?

Difficulty: no degree bound for a_i

Shorter Proofs

Problem

Given $f, f_1, \dots, f_s \in R[X]$, $N \in \mathbb{N}$

Compute $a_i \in \langle X \rangle$, $c_i \in R$ such that

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

if existent, else return FAILED.

Question Is this problem decidable?

Difficulty: no degree bound for a_i

Theorem The problem is decidable!

Minimal Representation

f has minimal representation with N terms iff f can be rewritten to 0 in N steps

Minimal Representation

f has minimal representation with N terms iff f can be rewritten to 0 in N steps

Theorem Let $f, f_1, \ldots, f_s \in R[X]$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

then $\deg(a_i f_i) \leq D := (N+1)\text{max}\deg(f, f_1, \ldots, f_s)$.

Minimal Representation

f has minimal representation with N terms iff f can be rewritten to 0 in N steps

Theorem Let $f, f_1, \ldots, f_s \in R[X]$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leq N} c_i a_i \cdot f_i,$$

then $\deg(a_i f_i) \leq D := (N+1)\max \deg(f, f_1, \ldots, f_s)$.

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_i$$
with unknown $c_i \in R$ and all $a_i, \in \langle X \rangle$ with $\deg(a_i f_i) \leq D$.
2. Look for solution of the resulting linear system with $\leq N$ nonzero coordinates.

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_i$$
with unknown $c_i \in R$ and all $a_i, \in \langle X \rangle$ with $\deg(a_i f_i) \leq D$.
2. Look for solution of the resulting linear system with $\leq N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system
- Step 2 is difficult but decidable
- The algorithm is not practical for non-trivial examples

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_i$$
with unknown $c_i \in R$ and all $a_i, \in \langle X \rangle$ with $\deg(a_i f_i) \leq D$.
2. Look for solution of the resulting linear system with $\leq N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system
- Step 2 is difficult but decidable
- The algorithm is not practical for non-trivial examples

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_i$$
with unknown $c_i \in R$ and all $a_i, \in \langle X \rangle$ with $\deg(a_i f_i) \leq D$.
2. Look for solution of the resulting linear system with $\leq N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system $\rightsquigarrow$ signatures to reduce search space
- Step 2 is difficult but decidable
- The algorithm is not practical for non-trivial examples

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_i$$
with unknown $c_i \in R$ and all $a_i, \in \langle X \rangle$ with $\deg(a_i f_i) \leq D$.
2. Look for solution of the resulting linear system with $\leq N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system $\rightsquigarrow$ signatures to reduce search space
- Step 2 is difficult but decidable $\rightsquigarrow$ linear programming to approx. solution
- The algorithm is not practical for non-trivial examples

CERTIFYING IDEAL MEMBERSHIP TESTS

Daniela Kaufmann

TU Wien, Austria

Dagstuhl Seminar 25231 "Certifying Algorithms for Automated Reasoning"
Schloss Dagstuhl, Wadern, Germany

June 2, 2025

TU WIEN Informatics

FWF Austrian Science Fund

A practical algorithm

Given	$f, f_1, \dots, f_s \in R[X]$, a representation α of f with degree $\leq D \in \mathbb{N}$
Compute	an ℓ_1-minimal representation of f with degree $\leq D$

A practical algorithm

Given $f, f_1, \ldots, f_s \in R[X]$, a representation α of f with degree $\leq D \in \mathbb{N}$
Compute an ℓ_1-minimal representation of f with degree $\leq D$

1. Compute Gröbner basis of Syz$(f_1, \ldots, f_s)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_1, \ldots, a_k f_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

A practical algorithm

Given $f, f_1, \dots, f_s \in R[X]$, a representation α of f with degree $\leq D \in \mathbb{N}$
Compute an ℓ_1-minimal representation of f with degree $\leq D$

1. Compute Gröbner basis of Syz$(f_1, \dots, f_s)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_1, \dots, a_k f_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but better behaviour in practice

A practical algorithm

Given $f, f_1, \ldots, f_s \in R[X]$, a representation α of f with degree $\leq D \in \mathbb{N}$
Compute an ℓ_1-minimal representation of f with degree $\leq D$

1. Compute Gröbner basis of Syz$(f_1, \ldots, f_s)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_1, \ldots, a_k f_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but better behaviour in practice

- In general, no guarantee for shortest repr., but good behaviour in practice
- Many examples are even totally unimodular $\rightsquigarrow$ algorithm is guaranteed to return a shortest representation

A practical algorithm

Given $f, f_1, \dots, f_s \in R[X]$, a representation α of f with degree $\leq D \in \mathbb{N}$
Compute an ℓ_1-minimal representation of f with degree $\leq D$

1. Compute Gröbner basis of Syz$(f_1, \dots, f_s)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_1, \dots, a_k f_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but better behaviour in practice
- In general, no guarantee for shortest repr., but good behaviour in practice
- Many examples are even totally unimodular $\rightsquigarrow$ algorithm is guaranteed to return a shortest representation