Sparse Polynomials
The Power of Vocabulary

Professor James Davenport
University of Bath
(visiting Waterloo)
Discussions with Carette (McMaster) & Giesbrecht/Roche (Waterloo) gratefully acknowledged

28 April 2009
The complexity of a problem can depend on the representations of the inputs, and outputs.
The complexity of a problem can depend on the representations of the inputs, and outputs. We are all familiar with simple examples: unary/binary; unsorted/sorted etc.
The complexity of a problem can depend on the representations of the inputs, and outputs. We are all familiar with simple examples: unary/binary; unsorted/sorted etc. What about polynomials?
The complexity of a problem can depend on the representations of the inputs, and outputs. We are all familiar with simple examples: unary/binary; unsorted/sorted etc.

What about polynomials? We are particularly interested in divisibility questions (gcd, factoring etc.).
(Univariate) polynomials: \( f = \sum_{i=0}^{n} a_i \) with \( H = \max |a_i| \)
(Univariate) polynomials: \( f = \sum_{i=0}^{n} a_i \) with \( H = \max |a_i| \)

dense \( \langle n, a_n, \ldots, a_0 \rangle \): \( \log_2 n + (n + 1) \log_2 H \).
(Univariate) polynomials: \( f = \sum_{i=0}^{n} a_i \) with \( H = \max |a_i| \)

- **dense** \( \langle n, a_n, \ldots, a_0 \rangle \): \( \log_2 n + (n + 1) \log_2 H \).

- **t-sparse** \( \langle t, \langle e_1, a_{e_1} \rangle, \ldots, \langle e_t, a_{e_t} \rangle \rangle \) with \( a_{e_i} \neq 0, e_i > e_{i+1} \):
  \( \log_2 t + t(\log_2 n + \log_2 H) \).
(Univariate) polynomials: \( f = \sum_{i=0}^{n} a_i \) with \( H = \max |a_i| \)

- **dense** \( \langle n, a_n, \ldots, a_0 \rangle \): \( \log_2 n + (n + 1) \log_2 H \).
- **t-sparse** \( \langle t, \langle e_1, a_{e_1} \rangle, \ldots, \langle e_t, a_{e_t} \rangle \rangle \) with \( a_{e_i} \neq 0, e_i > e_{i+1} \): 
  \( \log_2 t + t(\log_2 n + \log_2 H) \).
- **Factored** \( f = \prod_{j=1}^{k} f_j^{n_j} \), \( f_j \) square-free, relatively prime.
(Univariate) polynomials: $f = \sum_{i=0}^{n} a_i$ with $H = \max |a_i|$

dense $\langle n, a_n, \ldots, a_0 \rangle$: $\log_2 n + (n + 1) \log_2 H$.

t-sparse $\langle t, \langle e_1, a_{e_1} \rangle, \ldots, \langle e_t, a_{e_t} \rangle \rangle$ with $a_{e_i} \neq 0$, $e_i > e_{i+1}$: $\log_2 t + t(\log_2 n + \log_2 H)$.

Factored $f = \prod_{j=1}^{k} f_j^{n_j}$, $f_j$ square-free, relatively prime.
(there’s also “completely factored”)
(Univariate) polynomials: \( f = \sum_{i=0}^{n} a_i \) with \( H = \max |a_i| \)

dense \( \langle n, a_n, \ldots, a_0 \rangle \): \( \log_2 n + (n + 1) \log_2 H \).

t-sparse \( \langle t, \langle e_1, a_{e_1} \rangle, \ldots, \langle e_t, a_{e_t} \rangle \rangle \) with \( a_{e_i} \neq 0, e_i > e_{i+1} \):
\( \log_2 t + t(\log_2 n + \log_2 H) \).

Factored \( f = \prod_{j=1}^{k} f_j^{n_j} \), \( f_j \) square-free, relatively prime.
(there’s also “completely factored”)

SLP=DAG \( s_i = x \) or constant or \( s_j \otimes s_k \): \( j, k < i, \otimes \in \{+,-,\ast\} \).
(Univariate) polynomials:  \( f = \sum_{i=0}^{n} a_i \) with  \( H = \max |a_i| \)

dense  \( \langle n, a_n, \ldots, a_0 \rangle \):  \( \log_2 n + (n + 1) \log_2 H \).

t-sparse  \( \langle t, \langle e_1, a_{e_1} \rangle, \ldots, \langle e_t, a_{e_t} \rangle \rangle \) with  \( a_{e_i} \neq 0, \ e_i > e_{i+1} \):  \( \log_2 t + t(\log_2 n + \log_2 H) \).

Factored  \( f = \prod_{j=1}^{k} f_{n_j}^{n_j} \),  \( f_j \) square-free, relatively prime.  
(there’s also “completely factored”)

SLP=DAG  \( s_i = x \) or constant or  \( s_j \otimes s_k \):  \( j, k < i \),  \( \otimes \in \{+, -, \ast\} \).

Additive Complexity  What is the minimal number of  \( \pm \) needed to write  \( f \)?
Dense has its drawbacks.
Dense has its drawbacks.

Compute \((x^{1000000} - 1) (x^{1000000} + 1)\)?
Dense has its drawbacks.

Compute \((x^{1000000} - 1)(x^{1000000} + 1)\) ?

Certainly Sir: please wait a moment while I do 1,000,002,999,997 multiplications by zero.
In practice

**Dense** has its drawbacks.

Compute \((x^{1000000} - 1) (x^{1000000} + 1)\)?

Certainly Sir: please wait a moment while I do 1,000,002,999,997 multiplications by zero.

**Factored** is used by QEPCAD and other specialist systems.
In practice

**Dense** has its drawbacks.

Compute \((x^{1000000} - 1)(x^{1000000} + 1)\)?

Certainly Sir: please wait a moment while I do 1,000,002,999,997 multiplications by zero.

**Factored** is used by QEPCAD and other specialist systems.

**SLP** hasn’t caught on (and is harder than sparse)
In practice

**Dense** has its drawbacks.

Compute \((x^{1000000} - 1) (x^{1000000} + 1)\)?

Certainly Sir: please wait a moment while I do 1,000,002,999,997 multiplications by zero.

**Factored** is used by QEPCAD and other specialist systems.

**SLP** hasn’t caught on (and is harder than sparse)

**Additive Complexity** is really a theoretical tool
So that leaves Sparse, but ... 

- Books always recommend sparse polynomials.
Books always recommend sparse polynomials.

They discuss addition, cunning algorithms for multiplying \( t \) and \( u \)-term polynomials in \( O(tu \log \min(t, u)) \) operations,
So that leaves Sparse, but . . .

- Books always recommend sparse polynomials.
- They discuss addition, cunning algorithms for multiplying $t$- and $u$-term polynomials in $O(tu \log \min(t, u))$ operations, rather than the more obvious $O(tu(\log t + \log u))$, 
Books always recommend sparse polynomials. They discuss addition, cunning algorithms for multiplying $t$- and $u$-term polynomials in $O(tu \log \min(t, u))$ operations, rather than the more obvious $O(tu(\log t + \log u))$, then silently switch to dense models.
So that leaves Sparse, but . . .

- Books always recommend sparse polynomials.
- They discuss addition, cunning algorithms for multiplying $t$- and $u$-term polynomials in $O(tu \log \min(t, u))$ operations, rather than the more obvious $O(tu(\log t + \log u))$,
- then silently switch to dense models.
- Sparse “gets too difficult”.
Difficulty 1 — Factoring

$x^{pq} - 1 = (x - 1)(x^p - 1 + \cdots + 1)(x^q - 1 + \cdots + 1)(x^{pq} - p - q - 1 + \cdots - 1)$

and so knowing the degree of the factors is equivalent to factoring $n = pq$.

It's not enough to require that $n$ be given factored, since this problem can be "dressed up", e.g.

$x^{pq} + 2 - 2x^{pq} + x^2 - 2 = (x^2 - 2)(x^{pq} - 1)$. 

\[ x^{pq} - 1 = (x-1)(x^{p-1}+\cdots+1)(x^{q-1}+\cdots+1)(x^{pq-p-q-1}+\cdots-1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \).
\[ x^{pq} - 1 = (x - 1)(x^{p-1} + \cdots + 1)(x^{q-1} + \cdots + 1)(x^{pq-p-q-1} + \cdots - 1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \). It’s not enough to require that \( n \) be given factored, since this problem can be “dressed up”, e.g.

\[ x^{pq+2} - 2x^{pq} + x^2 - 2 \]
\[ x^{pq} - 1 = (x - 1)(x^{p-1} + \cdots + 1)(x^{q-1} + \cdots + 1)(x^{pq-p-q-1} + \cdots - 1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \). It’s not enough to require that \( n \) be given factored, since this problem can be “dressed up”, e.g.

\[ x^{pq+2} - 2x^{pq} + x^2 - 2 = (x^2 - 2)(x^{pq} - 1). \]
Dense polynomials $f$ whose square has fewer terms [Verdenius1949].
Dense polynomials \( f \) whose square has \textit{fewer} terms [Verdenius1949]. [CoppersmithDavenport1991] considered complete polynomials of degree 12 of the form:

\[
C := (1 + 2 x - 2 x^2 + 4 x^3 - 10 x^4 + 50 x^5 + 125 x^6) (1 + a x^6).
\]  
\( \text{(1)} \)

When \( a \) has any one of eight values, the square has only 12 terms.
Dense polynomials $f$ whose square has fewer terms [Verdenius1949]. [CoppersmithDavenport1991] considered complete polynomials of degree 12 of the form:

$$C := (1 + 2x - 2x^2 + 4x^3 - 10x^4 + 50x^5 + 125x^6) \cdot (1 + ax^6).$$  \hspace{1cm} (1)

When $a$ has any one of eight values, the square has only 12 terms. Subsequently proved optimal by [Abbott2002].
Dense polynomials $f$ whose square has fewer terms [Verdenius1949]. [CoppersmithDavenport1991] considered complete polynomials of degree 12 of the form:

$$C := (1 + 2x - 2x^2 + 4x^3 - 10x^4 + 50x^5 + 125x^6) (1 + ax^6).$$

When $a$ has any one of eight values, the square has only 12 terms. Subsequently proved optimal by [Abbott2002]. This construction can compound to make $\liminf_{n \to \infty} \frac{\#(f^2)}{\#(f)} = 0$. 
Dense polynomials \( f \) whose square has fewer terms [Verdenius1949]. [CoppersmithDavenport1991] considered complete polynomials of degree 12 of the form:

\[
C := \left(1 + 2x - 2x^2 + 4x^3 - 10x^4 + 50x^5 + 125x^6\right) \left(1 + ax^6\right). \tag{1}
\]

When \( a \) has any one of eight values, the square has only 12 terms. Subsequently proved optimal by [Abbott2002].

This construction can compound to make \( \liminf_{n \to \infty} \frac{\#(f^2)}{\#(f)} = 0 \).

So \( \limsup_{n \to \infty} \frac{\# \gcd(g, g')}{\#(g)} = \infty \) (\( g = f^2 \)).
Kinds of Polynomial

We will say that a polynomial is \textit{cyclotomic}, if all its roots are roots of unity.
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.

A polynomial $f$ is *self-reciprocal* if $f(x) = x^n f(1/x)$. 

\[2x^2 - 5x + 2 = (2x - 1)(x - 2)\]
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.

A polynomial $f$ is *self-reciprocal* if $f(x) = x^n f(1/x)$.

The product of self-reciprocals is self-reciprocal, but the converse is not true: $2x^2 - 5x + 2$
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.

A polynomial \( f \) is *self-reciprocal* if \( f(x) = x^n f(1/x) \).

The product of self-reciprocals is self-reciprocal, but the converse is not true: \( 2x^2 - 5x + 2 = (2x - 1)(x - 2) \).
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.
A polynomial $f$ is *self-reciprocal* if $f(x) = x^n f(1/x)$.
The product of self-reciprocals is self-reciprocal, but the converse is not true: $2x^2 - 5x + 2 = (2x - 1)(x - 2)$.
A monic integer non-self-reciprocal polynomial has a product of roots greater than $\text{Root0f}(\theta^3 - \theta - 1) \approx 1.324717957$ [Smyth1971] (in absolute value).
Kinds of Polynomial

We will say that a polynomial is *cyclotomic*, if all its roots are roots of unity. Many authors reserve this for irreducible polynomials, but we will explicitly say “irreducible” when we need to.

A polynomial \( f \) is *self-reciprocal* if \( f(x) = x^n f(1/x) \).

The product of self-reciprocals is self-reciprocal, but the converse is not true: 
\[
2x^2 - 5x + 2 = (2x - 1)(x - 2).
\]

A monic integer non-self-reciprocal polynomial has a product of roots greater than \( \text{RootOf}(\theta^3 - \theta - 1) \approx 1.324717957 \)

Therefore the number of them is bounded by \( \text{polynomial}(t, \log_2 H) \) (independent of \( n \)).
Let $\Phi_k$ be the $k$-th irreducible cyclotomic polynomial:

$$\Phi_k(x) = \prod_{\gcd(j,k)=1} (x - e^{2\pi ij/k}).$$
Let $\Phi_k$ be the $k$-th irreducible cyclotomic polynomial:

$$\Phi_k(x) = \prod_{\gcd(j,k)=1} (x - e^{2\pi ij/k}).$$

$\Phi_k$ has degree $\phi_k = |\{j < k \mid \gcd(j, k) = 1\}|.$
Let $\Phi_k$ be the $k$-th irreducible cyclotomic polynomial:

$$\Phi_k(x) = \prod_{\gcd(j,k)=1} (x - e^{2\pi ij/k}).$$

$\Phi_k$ has degree $\phi_k = |\{j < k \mid \gcd(j, k) = 1\}|$. 

$$c \frac{k \log \log k}{\log k} \leq \phi(k) < k.$$
Let $\Phi_k$ be the $k$-th irreducible cyclotomic polynomial:

$$\Phi_k(x) = \prod_{\gcd(j,k)=1} (x - e^{2\pi ij/k}).$$

$\Phi_k$ has degree $\phi_k = |\{j < k \mid \gcd(j, k) = 1\}|$.

$$c \frac{k \log \log k}{\log k} \leq \phi(k) < k.$$ 

In practice $\phi(k) > k/10$. 
Φ_k has surprising coefficient growth

<table>
<thead>
<tr>
<th>a_i</th>
<th>5 6 7 8=9 14 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Φ_k</td>
<td>1785 2805 3135 6545 10465 11305</td>
</tr>
<tr>
<td>φ(k)</td>
<td>768 1280 1440 3840 6336 6912</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a_i</th>
<th>25 27 59 359</th>
</tr>
</thead>
<tbody>
<tr>
<td>Φ_k</td>
<td>17225 20615 26565 40755</td>
</tr>
<tr>
<td>φ(k)</td>
<td>10752 12960 10560 17280</td>
</tr>
</tbody>
</table>
\( \Phi_k \) has surprising coefficient growth

Not always \( \pm 1 \).
$\Phi_k$ has surprising coefficient growth

Not always $\pm 1$. $\Phi_{105}$ has the first $\pm 2$, $\Phi_{385}$ the first $\pm 3$
\( \Phi_k \) has surprising coefficient growth

Not always \( \pm 1 \). \( \Phi_{105} \) has the first \( \pm 2 \), \( \Phi_{385} \) the first \( \pm 3 \) and \( \Phi_{1365} \) the first \( \pm 4 \).
Φ₁₀₅ has surprising coefficient growth Not always ±1. Φ₁₀₅ has the first ±2, Φ₃₈₅ the first ±3 and Φ₁₃₆₅ the first ±4. Thereafter the situation behaves as follows:

<table>
<thead>
<tr>
<th>Table: Large coefficients in Φₖ</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
</tr>
<tr>
<td>first ( \Phi_k )</td>
</tr>
<tr>
<td>( \phi(k) )</td>
</tr>
<tr>
<td>(</td>
</tr>
<tr>
<td>first ( \Phi_k )</td>
</tr>
<tr>
<td>( \phi(k) )</td>
</tr>
</tbody>
</table>
Cyclotomics mean that

- A sparse polynomial can have dense factors:
  \[ x^p - 1 = (x - 1)(x^{p-1} + \cdots + 1) \]
Cyclotomics mean that

- A sparse polynomial can have dense factors:
  \[ x^p - 1 = (x - 1)(x^{p-1} + \cdots + 1) \]
- The coefficients can be much larger than you would expect
Cyclotomics mean that

- A sparse polynomial can have dense factors:
  \[ x^p - 1 = (x - 1)(x^{p-1} + \cdots + 1) \]
- The coefficients can be much larger than you would expect
- The cofactors of the gcd of sparse polynomials can be dense:
  \[ \gcd(x^p - 1, x^q - 1) \]
Cyclotomics mean that

- A sparse polynomial can have dense factors:
  \[ x^p - 1 = (x - 1)(x^{p-1} + \cdots + 1) \]

- The coefficients can be much larger than you would expect

- The cofactors of the gcd of sparse polynomials can be dense:
  \[ \gcd(x^p - 1, x^q - 1); \]

- The square-free decomposition of sparse polynomials can be dense:
  \[ \text{sqfr}((x^p - 1)^2(x^q - 1)) = (x - 1)^3(x^{p-1} + \cdots + 1)^2(x^{q-1} + \cdots + 1). \]
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
2. Deciding is a sparse polynomial has a root of modulus 1.
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
2. Deciding if a sparse polynomial has a root of modulus 1.
   (Note this is \textit{not} quite the same as “has a cyclotomic factor”).
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
2. Deciding if a sparse polynomial has a root of modulus 1.
   (Note this is *not* quite the same as “has a cyclotomic factor”).
3. Determine whether $x^n - 1$ divides a given set of sparse polynomials (actually NP-complete).
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
2. Deciding if a sparse polynomial has a root of modulus 1.
   (Note this is *not* quite the same as “has a cyclotomic factor”).
3. Determine whether $x^n - 1$ divides a given set of sparse polynomials (actually NP-complete).
4. Various questions about quotients and remainders.
[Plaisted1984] has a number of NP-hardness results.

1. Deciding if two sparse polynomials are relatively prime.
2. Deciding is a sparse polynomial has a root of modulus 1.
   (Note this is not quite the same as “has a cyclotomic factor”).
3. Determine whether $x^n - 1$ divides a given set of sparse polynomials (actually NP-complete).
4. Various questions about quotients and remainders.
   These are reductions from 3-SAT, or from finding least primes in arithmetic progressions.
[Lenstra1999b] has a polynomial-time procedure that will find low-degree \( (\leq d) \) factors of a sparse polynomial: in fact polynomial\( (d,t,\log H) \) and independent of the input degree.
Re: Difficulty 1 — Factoring

\[ x^{pq-1} = (x-1)(x^{p-1} + \cdots + 1)(x^{q-1} + \cdots + 1)(x^{pq-p-q-1} + \cdots - 1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \). It's not enough to require that \( n \) be given factored, since this problem can be "dressed up", e.g.

\[ x^{pq+2} - 2x^{pq} + x^2 - 2 = (x^2 - 2)(x^{pq} - 1) \]

We assume an "integer factorization oracle", but it can't be called "too much":

\[ \sum \left\lfloor \log_2 k_i \right\rfloor \leq \log_2 n. \]
Re: Difficulty 1 — Factoring

\[ x^{pq} - 1 = (x - 1)(x^{p-1} + \cdots + 1)(x^{q-1} + \cdots + 1)(x^{pq-p-q-1} + \cdots - 1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \). It’s not enough to require that \( n \) be given factored, since this problem can be “dressed up”, e.g.

\[ x^{pq+2} - 2x^{pq} + x^2 - 2 = (x^2 - 2)(x^{pq} - 1). \]
\[ x^{pq} - 1 = (x - 1)(x^{p-1} + \cdots + 1)(x^{q-1} + \cdots + 1)(x^{pq-p-q-1} + \cdots - 1) \]

and so knowing the degree of the factors is equivalent to factoring \( n = pq \). It’s not enough to require that \( n \) be given factored, since this problem can be “dressed up”, e.g.

\[ x^{pq+2} - 2x^{pq} + x^2 - 2 = (x^2 - 2)(x^{pq} - 1). \]

We assume an “integer factorization oracle”, but it can’t be called “too much”: \( \sum \lfloor \log_2 k_i \rfloor \leq \log_2 n \).
Accept that most of our “common sense” bounds are wrong, and “common sense” estimates may be wrong.
Accept that most of our “common sense” bounds are wrong, and “common sense” estimates may be wrong. This is the hard part!
Accept that most of our “common sense” bounds are wrong, and “common sense” estimates may be wrong. This is the hard part!. Either produce procedures that will look for an answer, but not guarantee to find it, or resort to a reserve procedure.
Ignore them,
Ignore them,
i.e. produce algorithms for inputs which are guaranteed cyclotomic-free.
Ignore them, i.e. produce algorithms for inputs which are guaranteed cyclotomic-free.

Or at least detect them — hard in theory, easy in practice.
1. Ignore them, i.e. produce algorithms for inputs which are guaranteed cyclotomic-free.
2. Or at least detect them — hard in theory, easy in practice.
3. Or make them first-class citizens.
Cyclotomics as first-class citizens (1)

As well as an ordinary sparse polynomial, admit \( \Phi_k \) in the output, so that "factor \( x^p - 1 \) gives \( (x - 1)\Phi_p(x) \) as the output. Similarly \( \text{sqfr}((x^p - 1)^2(x^q - 1)) = (x - 1)^3\Phi_p(x)^2\Phi_q(x) \).

In order to answer questions like "what is the degree?", we probably need to attach the factorization of \( k \) to \( \Phi_k \).
As well as an ordinary sparse polynomial, admit $\Phi_k$ in the output, so that “factor $x^p - 1$” gives
As well as an ordinary sparse polynomial, admit $\Phi_k$ in the output, so that “factor $x^p - 1$” gives

$(x - 1)\Phi_p(x)$ as the output.

Similarly

$$\text{sqfr}((x^p - 1)^2(x^q - 1)) = (x - 1)^3\Phi_p(x)^2\Phi_q(x).$$
As well as an ordinary sparse polynomial, admit $\Phi_k$ in the output, so that "factor $x^p - 1$" gives $(x - 1)\Phi_p(x)$ as the output. Similarly

$$\text{sqfr}((x^p - 1)^2(x^q - 1)) = (x - 1)^3\Phi_p(x)^2\Phi_q(x).$$

In order to answer questions like "what is the degree?", we probably need to attach the factorization of $k$ to $\Phi_k$. 
As well as an ordinary sparse polynomial, admit $C_k = x^k - 1$ in the output, so that "factor $x^p - 1$" gives $C_p(x)$ as the output. Similarly $\text{sqfr}((x^p - 1)^2 (x^q - 1)) = C_p(x)^2 C_q(x)$. In order to answer questions like "how many factors are there?" or "what degree are they?", we probably need to attach the factorization of $k$ to $C_k$. 
As well as an ordinary sparse polynomial, admit $C_k = x^k - 1$ in the output, so that “factor $x^p - 1$” gives
As well as an ordinary sparse polynomial, admit $C_k = x^k - 1$ in the output, so that “factor $x^p - 1$” gives $C_p(x)$ as the output. Similarly

$$\text{sqfr}((x^p - 1)^2(x^q - 1)) = C_p(x)^2 C_q(x).$$
As well as an ordinary sparse polynomial, admit $C_k = x^k - 1$ in the output, so that “factor $x^p - 1$” gives $C_p(x)$ as the output. Similarly

$$\text{sqfr}((x^p - 1)^2(x^q - 1)) = C_p(x)^2C_q(x).$$

In order to answer questions like “how many factors are there?” or “what degree are they?”, we probably need to attach the factorization of $k$ to $C_k$. 
In theory, option 2 is preferable, but I’d advise a computer algebra system manufacturer to make option 1 the default.
In theory, option 2 is preferable, but I’d advise a computer algebra system manufacturer to make option 1 the default. In theory it makes no difference, but in practice I’d advise allowing “scaled cyclotomics” in the answer as well, to allow for the wise guy who asks “factor $x^{1000000} - 2^{1000000} = 2^{1000000} C_{1000000}(x/2)$”.
[Plaisted1984] has a number of NP-hardness results.
[Plaisted1984] has a number of NP-hardness results. Ignore them!
[Plaisted1984] has a number of NP-hardness results. Ignore them! (only joking)
This is to say, our algorithms might:

- occasionally take a very long time;
[Plaisted1984] has a number of NP-hardness results. Ignore them! (only joking)

This is to say, our algorithms might:

- *occasionally* take a very long time;
- *occasionally* return “I couldn’t find a gc.d./factorization/…., but I can’t prove there isn’t one”.
More open questions

1. How dense can the g.c.d. of sparse polynomials be?
More open questions

1. How dense can the g.c.d. of sparse polynomials be? (Both in theory and in practice)
1. How dense can the g.c.d. of sparse polynomials be? (Both in theory and in practice)
2. How dense can the highest-multiplicity square-free factor be?
More open questions

1. How dense can the g.c.d. of sparse polynomials be? (Both in theory and in practice)
2. How dense can the highest-multiplicity square-free factor be?
3. How hard is finding the number of factors
More open questions

1. How dense can the g.c.d. of sparse polynomials be? (Both in theory and in practice)
2. How dense can the highest-multiplicity square-free factor be?
3. How hard is finding the number of factors
More open questions

1. How dense can the g.c.d. of sparse polynomials be? (Both in theory and in practice)
2. How dense can the highest-multiplicity square-free factor be?
3. How hard is finding the number of factors (note that knowing that $n$ is the product of $k$ distinct primes, without knowing what they are, is sufficient here)?