Subresultants and Reduced Polynomial Remainder Sequences

GEORGE E. COLLINS*

Thomas J. Watson Research Center, Yorktown Heights, New York

ABSTRACT. Let $\mathcal{I}$ be an integral domain, $\mathcal{P}(\mathcal{I})$ the integral domain of polynomials over $\mathcal{I}$. Le $P, Q \in \mathcal{P}(\mathcal{I})$ with $m = \deg(P) \geq n = \deg(Q) > 0$. Let M be the matrix whose determinan defines the resultant of P and Q. Let M_{ij} be the submatrix of M obtained by deleting the las j rows of P coefficients, the last j rows of Q coefficients and the last $2j+1$ columns, exceptin column $m + n - i - j$ $(0 \leq i \leq j < n)$. The polynomial $R_j(x) = \sum_{i=0}^{j} \det(M_{ij})x^i$ is the j-t *subresultant* of P and Q, R_0 being the resultant. If $b = \mathcal{L}(Q)$, the leading coefficient of Q, ther exist uniquely $R, S \in \mathcal{P}(\mathcal{I})$ such that $b^{m-n+1}P = QS + R$ with $\deg(R) < n$; define $\overline{\mathcal{R}}(P, Q) = R$ Define $P_i \in \mathcal{P}(\mathfrak{F})$, $\mathfrak{F}$ the quotient field of $\mathcal{I}$, inductively: $P_1 = P$, $P_2 = Q$, $P_3 = \overline{\mathcal{R}}(P_1, P_2)$ $P_{i+2} = \overline{\mathcal{R}}(P_i, P_{i+1})/c_i^{\delta_{i-1}+1}$ for $i \geq 2$ and $n_{i+1} > 0$, where $c_i = \mathcal{L}(P_i)$, $n_i = \deg(P_i)$ an $\delta_i = n_i - n_{i+1}$. $P_1, P_2, \cdots, P_k$, for $k \geq 3$, is called a *reduced polynomial remainder sequence* Some of the main results are: (1) $P_i \in \mathcal{P}(\mathcal{I})$ for $1 \leq i \leq k$; (2) $P_k = \pm A_k R_{n_{k-1}-1}$, wher $A_k = \prod_{i=2}^{k-2} c_i^{\delta_{i-1}(\delta_i - 1)}$; (3) $c_k^{\delta_{k-1}-1} P_k = \pm A_{k+1} R_{n_k}$; (4) $R_j = 0$ for $n_k < j < n_{k-1} - 1$. Takin $\mathcal{I}$ to be the integers I, or $\mathcal{P}^r(I)$, these results provide new algorithms for computing resultant or greatest common divisors of univariate or multivariate polynomials. Theoretical analysi and extensive testing on a high-speed computer show the new g.c.d. algorithm to be faste than known algorithms by a large factor. When applied to bivariate polynomials, for example this factor grows rapidly with the degree and exceeds 100 in practical cases.

1. *Introduction*

Let $\mathcal{I}$ be an integral domain, $\mathcal{P}(\mathcal{I})$ the integral domain of polynomials with coefficients in $\mathcal{I}$. Small letters $a, b, c, \cdots$ are used for elements of $\mathcal{I}$, capital letters $P, Q, R, \cdots$ for elements of $\mathcal{P}(\mathcal{I})$ and $x, y, \cdots$ for variables. As is well known, if $\deg(P) \geq \deg(Q) > 0$, there exist $a \neq 0$, $b \neq 0$, S and R such that $aP = QS + bR$ and $\deg(R) < \deg(Q)$. Say that *R is a remainder of P modulo Q*. Two polynomials U and V are called *associates* in case there exist $c \neq 0$ and $d \neq 0$ such that $cU = dV$, and we write $U \sim V$. A remainder R is unique to within associates. In fact, more generally, if $P \sim P_1$, $Q \sim Q_1$ and R is a remainder of P modulo Q, then $R \sim R_1$ if and only if R_1 is a remainder of P_1 modulo Q_1.

We say $P_1, P_2, \cdots, P_k$ $(k \geq 3)$ is a *polynomial remainder sequence* (p.r.s.) if P_{i+2} is a remainder of P_i modulo P_{i+1} for $1 \leq i \leq k - 2$; it is *complete* in case $\deg(P_k) = 0$. The zero polynomial is assumed to have degree 0. Clearly, $\deg(P_1) \geq \deg(P_2) > \cdots > \deg(P_k) \geq 0$. For any P_1 and P_2 with $\deg(P_1) \geq \deg(P_2) > 0$, there exists a complete p.r.s. $P_1, P_2, \cdots, P_k$. If $P_1, P_2, \cdots, P_k$ and $Q_1, Q_2, \cdots, Q_r$ are complete p.r.s.'s with $P_1 \sim Q_1$ and $P_2 \sim Q_2$, then $k = r$ and $P_i \sim Q_i$ for $1 \leq i \leq k$.

For any $P \in \mathcal{P}(\mathcal{I})$, denote by $\mathcal{L}(P)$ the leading coefficient of P (if $P = 0$, we set $\mathcal{L}(P) = 0$). Let $m = \deg(P)$, $n = \deg(Q)$, $m \geq n > 0$, and define $\rho(P, Q) = R$, where $R(x) = \mathcal{L}(Q) \cdot P(x) - \mathcal{L}(P) \cdot x^{m-n} \cdot Q(x)$. Inductively, define $\rho^0(P, Q) = P$ and $\rho^{i+1}(P, Q) = \rho(\rho^i(P, Q), Q)$. Since $\deg(\rho(P, Q)) < \deg(P)$,

[*] Present address: Computer Sciences Department, University of Wisconsin, Madison, Wisconsin.

Journal of the Association for Computing Machinery, Vol. 14, No. 1, January 1967, pp. 128-142

there exists a least positive integer k such that $\deg(\rho^k(P, Q)) < \deg(Q)$. We call k the *rank of* P *over* Q and write $k = r(P, Q)$. Clearly, $r(P, Q) \leq m - n + 1$. Set $\mathfrak{R}(P, Q) = \rho^k(P, Q)$, where $k = r(P, Q)$; then, by induction on k, there exists S such that $\mathfrak{L}(Q)^k \cdot P = Q \cdot S + \mathfrak{R}(P, Q)$. Call $\mathfrak{R}(P, Q)$ the *Euclidean remainder of* P *modulo* Q. A p.r.s. $P_1, P_2, \cdots, P_k$ is *Euclidean* in case $P_{i+2} = \mathfrak{R}(P_i, P_{i+1})$ for $1 \leq i \leq k - 2$.

Let $P(x) = \sum_{i=0}^m a_i x^i$, $Q(x) = \sum_{i=0}^n b_i x^i$, and let M be the matrix whose determinant defines the resultant of P and Q; i.e., M is the following $m+n$ by $m+n$ matrix:

$$\begin{bmatrix}
a_m & a_{m-1} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_1 & a_0 & 0 & \cdot & \cdot & 0 & 0 \\
0 & a_m & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_2 & a_1 & a_0 & \cdot & \cdot & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & a_m & a_{m-1} & a_{m-2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_1 & a_0 \\
b_n & b_{n-1} & \cdot & \cdot & b_1 & b_0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & b_n & \cdot & \cdot & b_2 & b_1 & b_0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b_n & b_{n-1} & b_{n-2} & \cdot & \cdot & b_1 & b_0
\end{bmatrix}$$

M is called the *Sylvester matrix of* P *and* Q. Let M_{ij} be the submatrix of M obtained by deleting the last j of the n rows of P coefficients, the last j of the m rows of Q coefficients and the last $2j+1$ columns, excepting column $m + n - i - j$, for $0 \leq i \leq j < n$. The polynomial R_j, defined by $R_j(x) = \sum_{i=0}^j \det(M_{ij})x^i$, is called the *j-th subresultant of* P *and* Q, for $0 \leq j < n$. Notice that $\deg(R_j) \leq j$, and R_0 is the resultant of P and Q.

Let $P_1, P_2, \cdots, P_k$ be a Euclidean p.r.s. with $\deg(P_i) = n_i$. Let R_j be the jth subresultant of P_1 and P_2. In [1] it was shown that P_k is an associate of $R_{n_{k-1}-1}$, and, in fact, explicit expressions were obtained for elements a and b of $\mathfrak{I}$ such that $aP_k = \pm bR_{n_{k-1}-1}$, these expressions being products of powers of the leading coefficients of $P_1, P_2, \cdots, P_{k-1}$. A p.r.s. $P_1, P_2, \cdots, P_k$ is *regular* in case $r(P_i, P_{i+1}) = \deg(P_i) - \deg(P_{i+2})$ for $1 \leq i \leq k - 2$; it is *normal* in case $\deg(P_i) - \deg(P_{i+1}) = 1$ for $2 \leq i \leq k - 1$. From the definition of rank, we see that every regular p.r.s. is normal. We also obtained in [1] the corollary that if $P_1, P_2, \cdots, P_k$ is a regular Euclidean p.r.s., then $P_k = \pm cR_{n_{k-1}-1}$, where c is likewise explicitly given as a product of powers of the leading coefficients of $P_1, P_2, \cdots, P_{k-1}$.

These results established clearly that, excluding certain exceptional cases, the Euclidean algorithm in this form methodically introduces certain extraneous constant factors (i.e., powers of the leading coefficients) at a very rapid rate, and is therefore inefficient. The results also engendered some speculation in [1] as to how best to circumvent this source of inefficiency. Two possible methods were proposed, criticized and dismissed. Another method was suggested uncritically but apprehensively.

By methods of proof similar to those used in [1], we obtain in the present paper two new theorems on p.r.s. Theorem 1 pertains to what will be called a reduced p.r.s., this being a p.r.s. produced by a modification of the Euclidean algorithm of [1]. In this modification, we discard the notion of rank and in place of the function $\mathfrak{R}$, we use $\bar{\mathfrak{R}}$. We define $\bar{\mathfrak{R}}(P, Q)$ as the unique R such that, for some S, $\mathfrak{L}(Q)^{m-n+1} \cdot P = Q \cdot S + R$ and $\deg(R) < \deg(Q)$, where $m = \deg(P) \geq n = \deg(Q) > 0$. In addition to this change, we divide each remainder, $P_{i+2} = \bar{\mathfrak{R}}(P_i, P_{i+1})$, beginning with $i = 2$, by $\mathcal{L}(P_i)^{n_{i-1}-n_i+1}$. The theorem shows these divisions are always possible, and that the resulting p.r.s. bears a remarkably simple relationship to the sequence of subresultants. Indeed, $P_k = \pm dR_{n_{k-1}-1}$ and again d is explicitly given as a product of powers of the leading coefficients of $P_1, P_2, \cdots, P_{k-2}$; if $P_1, P_2, \cdots, P_k$ is normal, then $d = 1$.

As a corollary of Theorem 1, if $P_1, P_2, \cdots, P_k$ is any p.r.s., then P_i is an associate of $R_{n_{i-1}-1}$ for $i \geq 2$, and moreover every subresultant is either zero or is an associate of some $R_{n_{i-1}-1}$. We, therefore, set $S_1 = P_1$, $S_2 = P_2$ and $S_i = R_{n_{i-1}-1}$ for $i \geq 3$ and $n_{i-1} > 0$, and we call $S_1, S_2, S_3, \cdots, S_k$ a *subresultant p.r.s.* Theorem 1 provides an algorithm for computing the reduced p.r.s. $P_1, P_2, \cdots, P_k$ and from it the subresultant p.r.s. $S_1, S_2, \cdots, S_k$. Theorem 2 provides an algorithm for generating the subresultant p.r.s. directly. More specifically, one computes S_{i+2} as $\bar{\mathcal{R}}(S_i, S_{i+1}) \cdot a_i/b_i$, where a_i and b_i are products of powers of $\mathcal{L}(S_2), \cdots, \mathcal{L}(S_i)$.

The algorithms provided by these theorems and their corollaries for operations in $\mathcal{P}(\mathcal{I})$ are applicable whenever we have algorithms for operations in the integral domain $\mathcal{I}$. One significant example (indeed, the one which motivated the investigation) is obtained by taking $\mathcal{I}$ to be $\mathcal{I}_0$, the integral domain of the integers; or the integral domain, $\mathcal{I}_0[x_1, \cdots, x_n]$ (or, what is essentially the same, $\mathcal{P}^n(\mathcal{I}_0)$), of n-variable polynomials with integer coefficients. The algorithms are useful for computing resultants and greatest common divisors of polynomials with integer coefficients in any number of variables.

The g.c.d. (greatest common divisor) algorithm provided by Theorem 1 (the reduced p.r.s. algorithm) has been programmed for the IBM 7094 computer and applied to numerous polynomials in one and two variables. For comparison, three other polynomial g.c.d. algorithms were also programmed for the same computer and applied to the same polynomials. Of these three, one is the commonly used Euclidean algorithm, one is the ALPAK algorithm (used in the ALPAK system and described in [2]) and the third, the primitive p.r.s. algorithm, is a simplified but superior version of the ALPAK algorithm.

The results of these tests are reported in Section 3. In brief, the Euclidean algorithm is so inefficient that it is unusable except for univariate polynomials of degree 10 or less. Among the other three, the reduced p.r.s. algorithm is two to six times faster than the others for univariate polynomials. For bivariate polynomials, the situation is quite different. The reduced p.r.s. algorithm rapidly becomes 100 times faster than its competitors, the ratio increasing rapidly with the degrees of the polynomials.

2. *Theoretical Results*

Before starting Theorem 1, let us make more precise the definition of a reduced p.r.s. Recall from Section 1 the definition of $\bar{\mathcal{R}}(P, Q)$. Let $\mathcal{F}$ be the quotient field of $\mathcal{I}$. Let $P_1, P_2, \cdots, P_k$ be a p.r.s. with elements $P_i \in \mathcal{P}(\mathcal{F})$. $P_1, P_2, \cdots, P_k$ is said to be a *reduced p.r.s.* when $P_1, P_2 \in \mathcal{P}(\mathcal{I})$; $P_3 = \bar{\mathcal{R}}(P_1, P_2)$; and $P_{i+2} = \bar{\mathcal{R}}(P_i, P_{i+1})/c_i^{\delta_{i-1}+1}$ for $2 \leq i \leq k - 2$, where $c_i = \mathcal{L}(P_i)$, $n_i = \deg(P_i)$ and $\delta_i = n_i - n_{i+1}$. Actually we show (Corollary 1.1) that every element P_i of a reduced p.r.s. belongs to $\mathcal{P}(\mathcal{I})$, but this does not follow immediately from the definition and so we temporarily consider p.r.s. over $\mathcal{P}(\mathcal{F})$.

As an aid in proving Theorem 1, we now establish some conventions relating matrices and polynomials. Let A be any matrix with r rows and s columns, $r \leq s$. We define a function $\mathcal{A}$ such that $\mathcal{A}(A)$ is a polynomial P with $\deg(P) \leq s - r$. Let A_i be the square submatrix of A obtained by deleting all of the last $s - r + 1$ columns of A, excepting column $s-i$, for $0 \leq i \leq s - r$. Then P is the polynomial $P(x) = \sum_{i=0}^{s-r} \det(A_i) \cdot x^i$. We call $P = \mathcal{A}(A)$ the *associated polynomial of* A.

Now let M be the Sylvester matrix of the polynomials P and Q, with $\deg(P) = m$ and $\deg(Q) = n$. Let M_j be the submatrix obtained from M by deleting the last j rows of P, the last j rows of Q and the last j columns. M_j has $m + n - 2j$ rows and $m + n - j$ columns, and the result of deleting from M_j all of the last $j+1$ columns, excepting column $m + n - j - i$, is the submatrix M_{ij} which was used to define the jth subresultant, R_j, of P and Q. It follows that $R_j = \mathcal{A}(M_j)$.

If A is a matrix with 1 row and s columns, $A = (a_1, \cdots, a_s)$, then it can be seen that $\mathcal{A}(A) = P$, where $P(x) = \sum_{i=0}^{s-1} a_{s-i} x^i$. We are thus led to associate with any r by s matrix also the sequence of polynomials $(P_1, \cdots, P_r)$, where $P_i = \mathcal{A}(A_i)$, A_i being the ith row of A. We set $\mathcal{A}^*(A) = (P_1, \cdots, P_r)$. Of course, A is uniquely determined by $\mathcal{A}^*(A)$, given the number of columns in A. As an example, let I be the polynomial $I(x) = x$. Then the Sylvester matrix, M, of P and Q can be described by $\mathcal{A}^*(M) = (I^{n-1}P, I^{n-2}P, \cdots, P, I^{m-1}Q, I^{m-2}Q, \cdots, Q)$.

In the following we denote by $\mathcal{S}_j(P, Q)$ the jth subresultant of P and Q. We now prove the following lemma.

LEMMA 1. *Let $P_1, P_2, \cdots, P_k$ be a reduced p.r.s. Let $c_i = \mathcal{L}(P_i)$ and $n_i = \deg(P_i)$ for $1 \leq i \leq k$. Let $\delta_0 = -1$ and $\delta_i = n_i - n_{i+1}$ for $1 \leq i \leq k - 1$. Let $1 \leq i \leq k - 2$. Then,*

(*a*) $\mathcal{S}_j(P_i, P_{i+1}) = (-1)^{(n_i-j)(n_{i+1}-j)} \cdot c_i^{(\delta_{i-1}+1)(n_{i+1}-j)} \cdot c_{i+1}^{-(\delta_i+1)(n_{i+1}-j)+\delta_i+\delta_{i+1}}$
$\cdot \mathcal{S}_j(P_{i+1}, P_{i+2})$ *for* $0 \leq j < n_{i+2}$;

(*b*) $\mathcal{S}_j(P_i, P_{i+1}) = (-1)^{(\delta_i+1)\delta_{i+1}} \cdot c_i^{\delta_{i+1}(\delta_{i-1}+1)} \cdot c_{i+1}^{-\delta_i(\delta_{i+1}-1)} \cdot c_{i+2}^{\delta_{i+1}-1} \cdot P_{i+2}$ *for* $j = n_{i+2}$;

(*c*) $\mathcal{S}_j(P_i, P_{i+1}) = (-1)^{\delta_i+1} \cdot c_i^{\delta_{i-1}+1} \cdot P_{i+2}$ for $j = n_{i+1} - 1$;

(*d*) $\mathcal{S}_j(P_i, P_{i+1}) = 0$ *for* $n_{i+2} < j < n_{i+1} - 1$.

PROOF. Let M_j be the matrix with $\mathcal{A}^*(M_j) = (I^{n_{i+1}-j-1}P_i, I^{n_{i+1}-j-2}P_i, \cdots, P_i, I^{n_i-j-1}P_{i+1}, I^{n_i-j-2}P_{i+1}, \cdots, P_{i+1})$, so that $\mathcal{S}_j(P_i, P_{i+1}) = \mathcal{A}(M_j)$. By the definition of a reduced p.r.s., $c_{i+1}^{\delta_i+1} \cdot P_i = P_{i+1} \cdot Q_i + c_i^{\delta_{i-1}+1} \cdot P_{i+2}$ for some polynomial Q_i (including the case $i = 1$, since $\delta_0 = -1$). Since $n_{i+2} < n_i$, it follows that $n_i = \deg(P_{i+1} \cdot Q_i) = n_{i+1} + \deg(Q_i)$. Hence, $\deg(Q_i) = \delta_i$ and $P_{i+1} \cdot Q_i$ is a linear combination, with coefficients in $\mathcal{I}$, of $P_{i+1}, IP_{i+1}, \cdots, I^{\delta_i}P_{i+1}$. More generally,

$$c_{i+1}^{\delta_i+1} \cdot I^r P_i = I^r P_{i+1} \cdot Q_i + c_i^{\delta_{i-1}+1} I^r \cdot P_{i+2},$$

and $I^r P_{i+1} \cdot Q_i$ is a linear combination, with coefficients in $\mathcal{I}$, of $I^r P_{i+1}$, $I^{r+1}P_{i+1}$, $\cdots$, $I^{r+\delta_i}P_{i+1}$. All of these polynomials occur in $\mathcal{A}^*(M_j)$ provided $r \geq 0$ and $r + \delta_i \leq n_i - j - 1$, i.e., $0 \leq r \leq n_{i+1} - j - 1$. Hence, if we multiply each of the first $n_{i+1} - j$ rows of M_j by $c_{i+1}^{\delta_i+1}$ and subtract from each a suitable linear combination of the last $n_i - j$ rows, we obtain the matrix M_j' such that

$$\mathcal{A}^*(M_j') = (c_i^{\delta_{i-1}+1} I^{n_{i+1}-j-1} P_{i+2}, c_i^{\delta_{i-1}+1} I^{n_{i+1}-j-2} P_{i+2}, \cdots, c_i^{\delta_{i-1}+1} P_{i+2},$$
$$I^{n_i-j-1} P_{i+1}, \cdots, P_{i+1}).$$

If we now multiply each of the first $n_{i+1} - j$ rows of M_j' by $c_i^{-\delta_{i-1}-1}$, we obtain M_j''

with $\mathcal{A}^*(M_j'') = (I^{n_{i+1}-j-1}P_{i+2}, \cdots, P_{i+2}, I^{n_i-j-1}P_{i+1}, \cdots, P_{i+1})$. Now rearrange the rows of M_j'' to produce M_j''', where $\mathcal{A}^*(M_j''') = (I^{n_i-j-1}P_{i+1}, \cdots, P_{i+1}, I^{n_{i+1}-j-1}P_{i+2}, \cdots, P_{i+2})$. By the derivation of M_j''' from M_j, we have

$$(-1)^{(n_i-j)(n_{i+1}-j)}c_i^{-(\delta_{i-1}+1)(n_{i+1}-j)}c_{i+1}^{(\delta_i+1)(n_{i+1}-j)}\mathcal{A}(M_j) = \mathcal{A}(M_j'''). \qquad (1)$$

We now consider the four cases of the lemma. Suppose first that $j < n_{i+2}$. Let M_j^* be the matrix obtained from M_j''' by deleting the first $n_i - n_{i+2} = \delta_i + \delta_{i+1}$ rows and columns. Clearly $\mathcal{A}(M_j^*) = \mathcal{S}_j(P_{i+1}, P_{i+2})$ and $\mathcal{A}(M_j''') = c_{i+1}^{\delta_i+\delta_{i+1}} \cdot \mathcal{A}(M_j^*)$, since the first $\delta_i + \delta_{i+1}$ diagonal elements of M_j''' are c_{i+1} and only zeros occur below these diagonal elements. Combining this with (1), and using $\mathcal{A}(M_j) = \mathcal{S}_j(P_i, P_{i+1})$, we obtain (a).

Now assume $j = n_{i+2}$. Then

$$\deg (I^{n_{i+1}-j-1}P_{i+2}) = (n_{i+1} - j - 1) + n_{i+2} = n_{i+1} - 1,$$

and hence M_j''' is a triangular matrix whose first $n_i - j$ diagonal elements are c_{i+1} and whose other $n_{i+1} - j$ diagonal elements are c_{i+2}. Hence

$$\mathcal{A}(M_j''') = c_{i+1}^{n_i-j}c_{i+2}^{n_{i+1}-j-1}P_{i+2} = c_{i+1}^{\delta_i+\delta_{i+1}}c_{i+2}^{\delta_{i+1}-1}P_{i+2}.$$

Setting $j = n_{i+2}$ in (1), we have

$$(-1)^{(\delta_i+\delta_{i+1})\delta_{i+1}}c_i^{-(\delta_{i-1}+1)\delta_{i+1}}c_{i+1}^{(\delta_i+1)\delta_{i+1}}\mathcal{S}_j(P_i, P_{i+1}) = \mathcal{A}(M_j''') = c_{i+1}^{\delta_i+\delta_{i+1}}c_{i+2}^{\delta_{i+1}-1}P_{i+2}.$$

Since $(-1)^{(\delta_i+\delta_{i+1})\delta_{i+1}} = (-1)^{(\delta_i+1)\delta_{i+1}}$ and $\delta_i + \delta_{i+1} - (\delta_i + 1)\delta_{i+1} = -\delta_i(\delta_{i+1} - 1)$, we obtain (b).

Assume $j = n_{i+1} - 1$. Then $\mathcal{A}^*(M_j''') = (I^{n_i-j-1}P_{i+1}, \cdots, P_{i+1}, P_{i+2})$. Hence M_j''' is triangular and $\mathcal{A}(M_j''') = c_{i+1}^{\delta_i+1}P_{i+2}$. Setting $j = n_{i+1} - 1$ in (1), we have

$$(-1)^{\delta_i+1}c_i^{-(\delta_{i-1}+1)}c_{i+1}^{\delta_i+1}\mathcal{S}_j(P_i, P_{i+1}) = \mathcal{A}(M_j''') = c_{i+1}^{\delta_i+1}P_{i+2},$$

from which (c) is obtained.

Finally, assume $n_{i+2} < j < n_{i+1} - 1$. Then

$$\deg (I^{n_{i+1}-j-1}P_{i+2}) = n_{i+1} - j - 1 + n_{i+2} = (n_{i+1} - 1) + (n_{i+2} - j) < n_{i+1} - 1,$$

and $n_{i+1} - j > 1$. Hence, M_j''' is triangular and the $(n_i - j + 1)$-th diagonal element, which is not the last, is zero. So $\mathcal{A}(M_j''') = 0$. Then, by (1), $\mathcal{A}(M_j) = 0$, establishing (d).

LEMMA 2. *Let $P_1, P_2, \cdots, P_k$ be a reduced p.r.s. Let $c_i = \mathcal{L}(P_i)$, $n_i = deg(P_i)$ and $\delta_i = n_i - n_{i+1}$. Let $2 \le r \le k - 2$ and set $\sigma_r = \sum_{i=1}^{r-1} (n_i - j)(n_{i+1} - j)$, where $0 \le j < n_{k-1}$. Then*

$$\mathcal{S}_j(P_1, P_2) = (-1)^{\sigma_r}\left[\prod_{i=2}^{r} c_i^{-\delta_{i-1}(\delta_i-1)}\right] c_r^{-(\delta_{r-1}+1)(n_{r+1}-j)}\mathcal{S}_j(P_r, P_{r+1}).$$

PROOF. Setting $i = 1$ in Lemma 1 yields

$$\mathcal{S}_j(P_1, P_2) = (-1)^{(n_1-j)(n_2-j)} \cdot c_2^{-(\delta_1+1)(n_2-j)+\delta_1+\delta_2} \cdot \mathcal{S}_j(P_2, P_3).$$

Since $-(\delta_1 + 1)(n_2 - j) + \delta_1 + \delta_2 = -(\delta_1 + 1)\delta_2 - (\delta_1 + 1)(n_3 - j) + \delta_1 + \delta_2 = -\delta_1(\delta_2 - 1) - (\delta_1 + 1)(n_3 - j)$, this proves Lemma 2 for $r = 2$. Assume Lemma2 holds for r and that $r + 1 \leq k - 2$. Using Lemma 1 with $i = r$, we have

$$\begin{aligned}\mathcal{S}_j(P_1, P_2) &= (-1)^{\sigma_r}\left[\prod_{i=2}^{r} c_i^{-\delta_{i-1}(\delta_i - 1)}\right] c_r^{-(\delta_{r-1}+1)(n_{r+1}-j)}\mathcal{S}_j(P_r, P_{r+1}) \\ &= (-1)^{\sigma_r + (n_r - j)(n_{r+1}-j)} \cdot \left[\prod_{i=2}^{r} c_i^{-\delta_{i-1}(\delta_i - 1)}\right] \\ &\quad \cdot c_r^{-(\delta_{r-1}+1)(n_{r+1}-j)+(\delta_{r-1}+1)(n_{r+1}-j)} \\ &\quad \cdot c_{r+1}^{-(\delta_r+1)(n_{r+1}-j)+\delta_r+\delta_{r+1}} \cdot \mathcal{S}_j(P_{r+1}, P_{r+2}).\end{aligned}$$

Since

$$\sigma_r + (n_r - j)(n_{r+1} - j) = \sigma_{r+1}$$

and

$$\begin{aligned}-(\delta_r + 1)(n_{r+1} - j) + \delta_r + \delta_{r+1} &= -(\delta_r + 1)\delta_{r+1} - (\delta_r + 1)(n_{r+2} - j) \\ &\quad + \delta_r + \delta_{r+1} = -\delta_r(\delta_{r+1} - 1) - (\delta_r + 1)(n_{r+2} - j),\end{aligned}$$

this shows Lemma 2 holds for $r+1$. By induction, this completes the proof of Lemma 2.

Theorem 1. *Let $P_1, P_2, \cdots, P_k$ be a reduced p.r.s. Let $c_i = \mathcal{L}(P_i)$, $n_i = \deg(P_i)$ and $\delta_i = n_i - n_{i+1}$. Then,*

(a) $\mathcal{S}_{n_k}(P_1, P_2) = (-1)^{\sigma_k}[\prod_{i=2}^{k-1} c_i^{-\delta_{i-1}(\delta_i-1)}]c_k^{\delta_{k-1}-1}P_k$ *where* $\sigma_k = \sum_{i=1}^{k-1} n_i n_{i+1} + (n_1 + k)n_k$;

(b) $\mathcal{S}_{n_{k-1}-1}(P_1, P_2) = (-1)^{\tau_k}[\prod_{i=2}^{k-2} c_i^{-\delta_{i-1}(\delta_i-1)}]P_k$ *where* $\tau_k = \sum_{i=1}^{k-2} n_i n_{i+1} + (n_1 + k)(n_{k-1} + 1)$;

(c) $\mathcal{S}_j(P_1, P_2) = 0$ *for* $n_k < j < n_{k-1} - 1$.

Proof. If $k = 3$, these results follow by setting $i = 1$ in Lemma 1; hence we may assume $k \geq 4$. Setting $r = k - 2$ and $j = n_k$ in Lemma 2, we obtain

$$\mathcal{S}_{n_k}(P_1, P_2) = (-1)^{\alpha}\left[\prod_{i=2}^{k-2} c_i^{-\delta_{i-1}(\delta_i-1)}\right]c_{k-2}^{-(\delta_{k-3}+1)\delta_{k-1}}\mathcal{S}_{n_k}(P_{k-2}, P_{k-1}),$$

where

$$\alpha = \sum_{i=1}^{k-3}(n_i - n_k)(n_{i+1} - n_k).$$

Setting $i = k - 2$ in (b) of Lemma 1, we obtain

$$\mathcal{S}_{n_k}(P_{k-2}, P_{k-1}) = (-1)^{(\delta_{k-2}+1)\delta_{k-1}} c_{k-2}^{\delta_{k-1}(\delta_{k-3}+1)} c_{k-1}^{-\delta_{k-2}(\delta_{k-1}-1)} c_k^{\delta_{k-1}-1} P_k .$$

Combining these two equations results in

$$\mathcal{S}_{n_k}(P_1, P_2) = (-1)^{\alpha_1} \cdot \left[\prod_{i=2}^{k-1} c_i^{-\delta_{i-1}(\delta_i-1)}\right] c_k^{\delta_{k-1}-1}P_k,$$

where

$$\alpha_1 = \sum_{i=1}^{k-3}(n_i - n_k)(n_{i+1} - n_k) + (\delta_{k-2} + 1)\delta_{k-1}.$$

Using "$\equiv$" to denote congruence modulo two,

$$\begin{aligned}\alpha_1 &\equiv \sum_{i=1}^{k-3} (n_i - n_k)(n_{i+1} - n_k) + (\delta_{k-2} + \delta_{k-1})\delta_{k-1} \\ &\equiv \sum_{i=1}^{k-2} (n_i - n_k)(n_{i+1} - n_k) \\ &\equiv \sum_{i=1}^{k-2} n_i n_{i+1} + n_k \sum_{i=1}^{k-2} n_i + n_k \sum_{i=1}^{k-2} n_{i+1} + (k-2)n_k \\ &\equiv \sum_{i=1}^{k-2} n_i n_{i+1} + n_k(n_1 + n_{k-1} + k) = \sum_{i=1}^{k-1} n_i n_{i+1} + (n_1 + k)n_k = \sigma_k ,\end{aligned}$$

proving (a).

Setting $r = k - 2$ and $j = n_{k-1} - 1$ in Lemma 2, we have

$$\mathcal{S}_{n_{k-1}-1}(P_1, P_2) = (-1)^\beta \left[\prod_{i=1}^{k-2} c_i^{-\delta_{i-1}(\delta_i - 1)}\right] c_{k-2}^{-(\delta_{k-3}+1)} \mathcal{S}_{n_{k-1}-1}(P_{k-2}, P_{k-1}),$$

where

$$\beta = \sum_{i=1}^{k-3} (n_i - n_{k-1} + 1)(n_{i+1} - n_{k-1} + 1).$$

Setting $i = k - 2$ in (c) of Lemma 1, we have

$$\mathcal{S}_{n_{k-1}-1}(P_{k-2}, P_{k-1}) = (-1)^{\delta_{k-2}+1} c_{k-2}^{\delta_{k-3}+1} P_k .$$

Combining these two equations results in (b) except with

$$\beta_1 = \sum_{i=1}^{k-3} (n_i - n_{k-1} + 1)(n_{i+1} - n_{k-1} + 1) + \delta_{k-2} + 1$$

in place of τ_k. However,

$$\begin{aligned}\beta_1 &\equiv \sum_{i=1}^{k-3} n_i n_{i+1} + (n_{k-1} + 1)(n_1 + n_{k-2}) + (k-3)(n_{k-1} + 1) + n_{k-2} + n_{k-1} + 1 \\ &\equiv \sum_{i=1}^{k-2} n_i n_{i+1} + n_{k-1} n_1 + n_1 + n_{k-2} + (k+1)(n_{k-1} + 1) + n_{k-2} + (n_{k-1} + 1) \\ &\equiv \sum_{i=1}^{k-2} n_i n_{i+1} + (n_{k-1} + 1)(n_1 + k + 1 + 1) \equiv \tau_k ,\end{aligned}$$

proving (b).

Assume $n_k < j < n_{k-1} - 1$. By Lemma 2, with $r = k - 2$, $\mathcal{S}_j(P_1, P_2) \sim \mathcal{S}_j(P_{k-2}, P_{k-1})$ and by (d) of Lemma 1, with $i = k - 2$, $\mathcal{S}_j(P_{k-2}, P_{k-1}) = 0$, proving (c).

COROLLARY 1.1. *Let $P_1, P_2, \cdots, P_k$ be a reduced p.r.s. Then $P_i \in \mathcal{P}(\mathcal{I})$ for $1 \le i \le k$.*

PROOF. By induction on k. For $k = 3$, the corollary holds by the definition of a reduced p.r.s. Assume it holds for k, and let $P_1, P_2, \cdots, P_{k+1}$ be a reduced p.r.s. Then $P_1, P_2, \cdots, P_k$ is a reduced p.r.s. so, by hypothesis, $P_1, P_2, \cdots, P_k \in \mathcal{P}(\mathcal{I})$. Hence, $c_1, c_2, \cdots, c_k \in \mathcal{I}$. By (b) of Theorem 1,

$$P_{k+1} = \pm \left[\prod_{i=2}^{k-1} c_i^{\delta_{i-1}(\delta_i - 1)}\right] \mathcal{S}_{n_k - 1}(P_1, P_2).$$

But $\mathcal{S}_{n_k-1} \in \mathcal{P}(\mathcal{I})$, and $\delta_{i-1}(\delta_i - 1) \geq 0$ for $2 \leq i \leq k - 1$. Hence $P_{k+1} \in \mathcal{P}(\mathcal{I})$.

Corollary 1.2. *Let $P_1, P_2, \cdots, P_k$ be a complete reduced p.r.s. Then every nonzero subresultant of P_1 and P_2 is an associate of some P_i.*

Proof. Let $\mathcal{S}_j(P_1, P_2)$ be any nonzero subresultant. Then $n_k = 0 \leq j < n_2$. Hence, for some i, $3 \leq i \leq k$, $n_i \leq j < n_{i-1}$. Now apply Theorem 1 with $k = i$. By (c) of Theorem 1, $j = n_i$ or $j = n_{i-1} - 1$, since $\mathcal{S}_j(P_1, P_2) \neq 0$. Hence, by (a) and (b) of Theorem 1, $\mathcal{S}_j(P_1, P_2) \sim P_i$.

Corollary 1.3. *Let $P_1, P_2, \cdots, P_k$ be a normal reduced p.r.s. Let $P_1, P_2, S_3, \cdots, S_k$ be a subresultant p.r.s. Let $n_i = deg\,(P_i)$. Then $P_k = (-1)^{(n_1+n_2+1)k} S_k$.*

Proof. By definition of a subresultant p.r.s., $S_k = \mathcal{S}_{n_{k-1}-1}(P_1, P_2)$. By normality, $\delta_i = n_i - n_{i+1} = 1$ for $1 < i \leq k - 1$. Now apply (b) of Theorem 1, noting that $n_i n_{i+1}$ is even for $1 < i \leq k - 2$ and that $n_{k-1} = n_2 - (k - 3)$.

Corollary 1.4. *Let $P_1, P_2, \cdots, P_k$ be a normal complete reduced p.r.s. Then P_k is the resultant of P_1 and P_2.*

Proof. Apply (a) of Theorem 1, noting that each $\delta_i = 1$, each $n_i n_{i+1}$ is even and $n_k = 0$.

Theorem 1, part (b), provides an algorithm for computing any term, S_k, of a subresultant p.r.s. $P_1, P_2, S_3, \cdots, S_k, \cdots$. Namely, one may compute the reduced p.r.s. $P_1, P_2, \cdots, P_k$ and then divide P_k by $(-1)^{\tau_k} \cdot \prod_{i=2}^{k-2} c_i^{-\delta_{i-1}(\delta_i - 1)}$. However, we now seek to obtain a direct method for computing the subresultant p.r.s., i.e., a method which provides a formula for S_k in terms of $P_1, P_2, S_3, \cdots, S_{k-1}$ without recourse to the reduced p.r.s. To this end the following lemma is first proved.

Lemma 3. *Let $P_1, P_2, \cdots, P_k$ be a p.r.s. in $\mathcal{P}(\mathfrak{F})$. Let $c_i = \mathcal{L}(P_i)$, $n_i = deg\,(P_i)$, $\delta_i = n_i - n_{i+1}$. Let $P_3 = (-1)^{e_1} \cdot \bar{\mathcal{R}}(P_1, P_2)$ and*

$$P_{i+2} = (-1)^{e_i} \cdot c_i^{-\delta_{i-1}-1} \cdot \left[\prod_{j=2}^{i} c_j^{f_{ij}}\right] \cdot \bar{\mathcal{R}}(P_i, P_{i+1}),$$

for $2 \leq i \leq k - 2$, where the e_i and f_{ij} are arbitrary integers. Let $P_1, P_2, S_3, \cdots, S_k$ be a subresultant p.r.s. Let

$$g_k = \sum_{i=1}^{k-2} n_i n_{i+1} + (n_1 + k)(n_{k-1} - 1) + \sum_{i=1}^{k-2} e_i(n_{i+1} - n_{k-1} + 1)$$

and

$$h_{ik} = \delta_{i-1}(\delta_i - 1) + \sum_{j=i}^{k-2} f_{ji}(n_{j+1} - n_{k-1} + 1).$$

Then

$$P_k = (-1)^{g_k} \cdot \left[\prod_{i=2}^{k-2} c_i^{h_{ik}}\right] \cdot S_k.$$

Proof. By generalizing the proof of Lemma 1 to apply to the p.r.s. $P_1, P_2, \cdots, P_k$ of the present lemma rather than the reduced p.r.s., we obtain in place of (a) of Lemma 1 the following:

$$\begin{aligned} \mathcal{S}_j(P_i, P_{i+1}) = (-1)^{(n_i-j)(n_{i+1}-j)+e_i(n_{i+1}-j)} & \\ \cdot c_i^{(\delta_{i-1}+1)(n_{i+1}-j)} \cdot c_{i+1}^{-(\delta_i+1)(n_{i+1}-j)+\delta_i+\delta_{i+1}} & \qquad (2) \\ \cdot \left[\prod_{p=2}^{i} c_p^{f_{ip}}\right]^{-(n_{i+1}-j)} \cdot \mathcal{S}_j(P_{i+1}, P_{i+2}) \text{ for } j < n_{i+2}. & \end{aligned}$$

Similarly, in place of (c) we obtain:

$$\mathcal{S}_j(P_i\,,\,P_{i+1}) = (-1)^{\delta_i+1+e_i}\cdot c_i^{\delta_{i-1}+1}\cdot\left[\prod_{p=2}^{i} c_p^{f_{ip}}\right]^{-1}\cdot P_{i+2} \quad \text{for} \quad j = n_{i+1} - 1. \tag{3}$$

Now we carry out an induction as in Lemma 2 and obtain:

$$\begin{aligned}\mathcal{S}_j(P_1\,,P_2) = (-1)^{\sigma_r}\cdot&\left[\prod_{i=2}^{r} c_i^{-\delta_{i-1}(\delta_i-1)}\right]\cdot c_r^{-(\delta_{r-1}+1)(n_{r+1}-j)}\\ &\cdot\left[\prod_{i=2}^{r-1} c_i^{-\Sigma_{p=i}^{r-1} f_{pi}(n_{p+1}-j)}\right]\cdot\mathcal{S}_j(P_r\,,P_{r+1})\end{aligned} \tag{4}$$

$$\text{for} \quad 2 \le r \le k-2 \quad \text{and} \quad j < n_{k-1}\,,$$

where

$$\sigma_r = \sum_{i=1}^{r-1}(n_i - j)(n_{i+1} - j) + \sum_{i=1}^{r-1} e_i(n_{i+1} - j).$$

As in the proof of (b) of Theorem 1, we now set $i = k - 2$ in (3) and $j = n_{k-1} - 1$ and $r = k - 2$ in (4), obtaining:

$$\mathcal{S}_{n_{k-1}-1}(P_{k-2}\,,P_{k-1}) = (-1)^{\delta_{k-2}+1+e_{k-2}}\cdot c_{k-2}^{\delta_{k-3}+1}\cdot\left[\prod_{i=2}^{k-2} c_i^{f_{k-2,i}}\right]^{-1}\cdot P_k\,; \tag{5}$$

$$\begin{aligned}S_k = (-1)^{\sigma_{k-2}}\cdot&\left[\prod_{i=2}^{k-2} c_i^{-\delta_{i-1}(\delta_i-1)}\right]\cdot c_{k-2}^{-(\delta_{k-3}+1)}\\ &\cdot\left[\prod_{i=2}^{k-3} c_i^{-\Sigma_{j=i}^{k-3} f_{ji}(n_{j+1}-n_{k-1}+1)}\right]\cdot\mathcal{S}_{n_{k-1}-1}(P_{k-2}\,,P_{k-1}),\end{aligned} \tag{6}$$

where

$$\sigma_{k-2} = \sum_{i=1}^{k-3}(n_i - n_{k-1} + 1)(n_{i+1} - n_{k-1} + 1) + \sum_{i=1}^{k-3} e_i(n_{i+1} - n_{k-1} + 1).$$

Combining (5) and (6) and simplifying the exponent of -1, modulo 2, yields the conclusion of Lemma 3.

We now seek to so determine the e_i and f_{ij} of Lemma 3 in such a way that the p.r.s. $P_1\,, P_2\,, \cdots, P_k$ of Lemma 3 coincides with the reduced p.r.s. $P_1\,, P_2\,, S_3\,, \cdots, S_k$, if possible. A sufficient condition for this coincidence is, by Lemma 3, that

$$\begin{aligned}g_r = \sum_{i=1}^{r-2} n_i n_{i+1} &+ (n_1 + r)(n_{r-1} - 1)\\ &+ \sum_{i=1}^{r-2} e_i(n_{i+1} - n_{r-1} + 1) \equiv 0 (\text{mod } 2) \quad \text{for} \quad 3 \le r \le k\end{aligned}$$

and

$$h_{ir} = \delta_{i-1}(\delta_i - 1) + \sum_{j=i}^{r-2} f_{ji}(n_{j+1} - n_{r-1} + 1) = 0 \quad \text{for} \quad 2 \le i \le r - 2 \le k - 2.$$

Setting $g_3 \equiv 0$ we have $n_1n_2 + (n_1 + 3)(n_2 - 1) + e_1(n_2 - n_2 + 1) \equiv 0$ and hence $e_1 \equiv n_1 + n_2 + 1 \equiv \delta_1 + 1$. Now we notice that $g_{r+1} - g_r \equiv n_{r-1}n_r + n_1\delta_{r-1}$

$+ r\delta_{r-1} + n_r + 1 + \sum_{i=1}^{r-2} e_i\delta_{r-1} + e_{r-1}$. Hence if $g_r \equiv g_{r+1} \equiv 0$ then $e_{r-1} \equiv \delta_{r-1}(\sum_{i=1}^{r-2} e_i + n_1 + n_r + r) + 1$. Setting $r = 3$ in this formula, we obtain $e_2 \equiv \delta_2(\delta_1 + 1 + n_1 + n_3 + 3) + 1 \equiv \delta_2(n_2 + n_3) + 1 \equiv \delta_2^2 + 1 \equiv \delta_2 + 1$. Similarly, setting $r = 4$ we obtain $e_3 \equiv \delta_3 + 1$. Set $e_i = \delta_i + 1$ for all i. Then

$$\begin{aligned} g_r &= (n_1 + r)(n_{r-1} - 1) + \sum_{i=1}^{r-2} [n_i n_{i+1} + (\delta_i + 1)(n_{i+1} - n_{r-1} + 1)] \\ &\equiv (n_1 + r)(n_{r-1} + 1) + \sum_{i=1}^{r-2} [n_i(n_{r-1} + 1) + (n_{i+1} + 1)(n_{i+1} + n_{r-1} + 1)] \\ &\equiv (n_1 + r)(n_{r-1} + 1) + \sum_{i=1}^{r-2} (n_i + n_{i+1} + 1)(n_{r-1} + 1) \\ &\equiv (n_{r-1} + 1)\left[(n_1 + r) + \sum_{i=1}^{r-2} (n_i + n_{i+1} + 1)\right] \\ &\equiv (n_{r-1} + 1)(n_1 + r + n_1 + n_{r-1} + r) \equiv (n_{r-1} + 1)n_{r-1} \equiv 0. \end{aligned}$$

We now make a similar determination of the f_{ij}. Setting $r = i + 2$, we obtain $h_{i,i+2} = \delta_{i-1}(\delta_i - 1) + f_{ii}$. Hence $h_{i,i+2} = 0$ implies $f_{ii} = -\delta_{i-1}(\delta_i - 1)$. Setting $r = i + 3$, we obtain $h_{i,i+3} = \delta_{i-1}(\delta_i - 1) + f_{ii}(\delta_{i+1} + 1) + f_{i+1,i}$. Hence $h_{i,i+3} = h_{i,i+2} = 0$ implies $f_{i+1,i} = -\delta_{i-1}(\delta_i - 1) + \delta_{i-1}(\delta_i - 1)(\delta_{i+1} + 1) = \delta_{i-1}(\delta_i - 1)\delta_{i+1}$. Similarly, setting $r = i + 4$, we obtain $\delta_{i-1}(\delta_i - 1) + f_{ii}(\delta_{i+1} + \delta_{i+2} + 1) + f_{i+1,i}(\delta_{i+2} + 1) + f_{i+2,i} = 0$,

$$\begin{aligned} f_{i+2,i} &= -\delta_{i-1}(\delta_i - 1) + \delta_{i-1}(\delta_i - 1)(\delta_{i+1} + \delta_{i+2} + 1) - \delta_{i-1}(\delta_i - 1)\delta_{i+1}(\delta_{i+2} + 1) \\ &= \delta_{i-1}(\delta_i - 1)(-1 + \delta_{i+1} + \delta_{i+2} + 1 - \delta_{i+1}\delta_{i+2} - \delta_{i+1}) \\ &= -\delta_{i-1}(\delta_i - 1)(\delta_{i+1} - 1)\delta_{i+2}. \end{aligned}$$

Assume $f_{ii} = -\delta_{i-1}(\delta_i - 1)$ and $f_{i+m,i} = (-1)^{m+1}\delta_{i-1} \cdot [\prod_{j=i}^{i+m-1} (\delta_j - 1)] \cdot \delta_{i+m}$. Then, as shown above, $h_{i,i+2} = 0$. Assume $h_{ir} = 0$ and $r \geq i + 2$. Then

$$h_{i,r+1} = h_{ir} + \delta_{r-1} \sum_{j=i}^{r-2} f_{ji} + f_{r-1,i} = \delta_{r-1} \sum_{j=i}^{r-2} f_{ji} + f_{r-1,i}.$$

However, one can easily show by induction that

$$\delta_{r-1} \sum_{j=i}^{r-2} f_{ji} + f_{r-1,i} = 0.$$

Hence $h_{i,r+1} = 0$ and, by induction, $h_{ir} = 0$ for all $r \geq i + 2$.

This completes the proof of the following theorem.

THEOREM 2. *Let $S_1, S_2, \cdots, S_k$ be a subresultant p.r.s. Let $c_i = \mathcal{L}(S_i)$, $n_i = deg\ (S_i)$, $\delta_i = n_i - n_{i+1}$. Set $f_{ii} = -\delta_{i-1}(\delta_i - 1)$ and $f_{i+r,i} = (-1)^{r+1}\delta_{i-1} \cdot [\prod_{j=i}^{i+r-1} (\delta_j - 1)] \cdot \delta_{i+r}$, for $i \geq 2$, $r \geq 1$ and $i + r \leq k - 2$. Then*

$$S_3 = (-1)^{\delta_1+1}\bar{\mathfrak{R}}(S_1, S_2),$$

and

$$S_{i+2} = (-1)^{\delta_i+1}\left[\prod_{j=2}^{i} c_j^{f_{ij}}\right] \cdot c_i^{-\delta_{i-1}-1} \cdot \bar{\mathfrak{R}}(S_i, S_{i+1})$$

for $2 \leq i \leq k - 2$.

3. *G.C.D. and Resultant Algorithms*

As already mentioned, Theorem 1 provides new algorithms for computing resultants and greatest common divisors of polynomials in any number of variables with coefficients from any integral domain $\mathcal{I}_0$ provided, of course, that we have available algorithms for the arithmetic operations in $\mathcal{I}_0$. If $r+1$ is the number of variables, then in applying Theorem 1 we rewrite our given polynomials as univariate polynomials in one variable with coefficients which are r-variate polynomials and take $\mathcal{I} = \mathcal{P}^r(\mathcal{I}_0)$ in Theorem 1. Although other choices are undoubtedly of interest, in the present paper we consider (with the exception of a few remarks) only the case where $\mathcal{I}_0$ is the integral domain of the integers.

First, consider algorithms for g.c.d. calculation. Let $P(x_1, \cdots, x_r, y)$ be a polynomial with integer coefficients in $r+1$ variables, $r \geq 0$. Let

$$P(x_1, \cdots, x_r, y) = \sum_{i=0}^{n} A_i(x_1, \cdots, x_r)\cdot y^i.$$

Applying induction on r, we may assume that we can compute $A = \gcd(A_0, A_1, \cdots, A_n)$, since for $r = 0$ we have the familiar Euclidean algorithm for computing the g.c.d. of integers. A is the *content* (with respect to y) and we write $A = \text{cont}\,(P)$. We can then compute $\bar{P} = P/A$. We call $\bar{P}$ the *primitive part* of P (with respect to y) and write $\bar{P} = pp(P)$. $\bar{P}$ is *primitive* (with respect to y), i.e., any common divisor of its coefficients $\bar{A}_0, \bar{A}_1, \cdots, \bar{A}_n$ ($\bar{A}_i = A_i/A$) is a unit of $\mathcal{I}_0[x_1, \cdots, x_r] = \mathcal{P}^r(\mathcal{I}_0)$. Of course, the only units of $\mathcal{P}^r(\mathcal{I}_0)$ are 1 and -1.

Now let Q_1, Q_2 be nonzero elements of $\mathcal{P}^{r+1}(\mathcal{I}_0)$ and suppose we wish to compute $Q = \gcd(Q_1, Q_2)$. First compute $A_i = \text{cont}\,(Q_i)$ and $P_i = pp(Q_i)$, for $i = 1, 2$; then $A = \gcd(A_1, A_2)$. It now suffices to compute $P = \gcd(P_1, P_2)$ since $Q = A\cdot P$, and we know that P is primitive. Let $n_i = \deg(P_i)$. Since $\gcd(P_1, P_2) = \gcd(P_2, P_1)$, we may assume $n_1 \geq n_2$. We may also assume $n_2 > 0$ since otherwise $P = 1$. Now let $P_1, P_2, \cdots, P_k$ be any complete p.r.s. The standard proof (see, for example, [3, Ch. XVI]) easily generalizes to show that $P = 1$ if $P_k \neq 0$, and $P = pp(P_{k-1})$ if $P_k = 0$.

We thus obtain, for each specification of an algorithm for computing a complete p.r.s. $P_1, P_2, \cdots, P_k$ starting with given primitive polynomials P_1 and P_2, a g.c.d. algorithm. We now consider four such g.c.d. algorithms. Perhaps the most natural, most obvious and most commonly used algorithm is the *Euclidean algorithm*, obtained by taking $P_1, P_2, \cdots, P_k$ to be the Euclidean p.r.s., generated according to its definition.

Now let $P_1, P_2, \cdots, P_k$ be a *primitive p.r.s.* which begins with P_1, P_2, i.e., a p.r.s. in which each P_i is primitive. We distinguish two algorithms depending on how such a p.r.s. is generated. The simplest generation method is given by $P_{i+2} = pp(\mathcal{R}(P_i, P_{i+1}))$. We call the resulting g.c.d. algorithm the *primitive p.r.s. algorithm.*

In the ALPAK system [2], the successive terms of a primitive p.r.s. $P_1, P_2, \cdots, P_k$ are generated in the following more complex way. The operation ρ of Section 1 is replaced by an operation $\bar{\rho}$. Let $m = \deg(P) \geq n = \deg(Q) > 0$, $a = \mathcal{L}(P)$, $b = \mathcal{L}(Q)$. Let $c = \gcd(a, b)$, $\bar{a} = a/c$, $\bar{b} = b/c$. Then $\bar{\rho}(P, Q) = pp(\bar{b}P - \bar{a}I^{m-n}Q)$. Now define $\bar{\rho}^0(P, Q) = P$ and, inductively, $\bar{\rho}^{i+1}(P, Q) = \bar{\rho}(\bar{\rho}^i(P, Q), Q)$. Then

$P_{i+2} = \bar{\rho}^k(P_i, P_{i+1})$, where $k = r(P_i, P_{i+1})$, and this describes the method used in ALPAK. The g.c.d. algorithm so obtained will be called the ALPAK algorithm.

The fourth g.c.d. algorithm is obtained by taking $P_1, P_2, \cdots, P_k$ to be the reduced p.r.s. which starts with P_1, P_2, generated according to its definition, and we call this the *reduced p.r.s. algorithm.*

Each of these four algorithms has been programmed for the IBM 7094 computer, within the framework of the PM polynomial manipulation system [4], and hence their implementations do not differ in any essential details. Each algorithm was applied to a set of 7 pairs $(P_1, Q_1), \cdots, (P_k, Q_k)$ of univariate polynomials and a set of 5 pairs $(R_3, S_3), \cdots, (R_7, S_7)$ of bivariate polynomials, each algorithm being applied to the same polynomials (except that some algorithms were too inefficient to do some problems in a reasonable amount of time). Each P_k, and each Q_k, is a polynomial of degree $5k$ with random integer coefficients of two decimal digits, i.e., chosen at random (with uniform distribution) from the set $\{n: |n| \leq 99\}$. Each R_k, and each S_k, is a polynomial of total degree k with random one-decimal-digit coefficients. Thus $R_k(x, y) = \sum_{i+j=0}^{k} a_{ij}x^iy^j$, with $a_{ij} \in \{n: |n| \leq 9\}$, has $(k + 1)(k + 2)/2$ terms (a few of which may happen to be zero). Not surprisingly, each pair proved to be relatively prime.

Table 1 gives the computing time required by each algorithm to compute the g.c.d. of each pair of polynomials to which it was applied.

We now add a few remarks tending to explain and interpret these results as well as to provide a basis for extrapolation.

Assume the Euclidean algorithm is applied to two univariate polynomials P_1 and P_2, both of degree n, whose coefficients are approximately d decimal digits long, and assume further that the Euclidean p.r.s. $P_1, P_2, \cdots, P_k$ is regular (which experience shows to be the typical case). Then the coefficients of P_3 will be approximately $2d$ digits long, those of P_4 approximately $5d$ digits long (since $r(P_2, P_3) = 2$), those of P_5 $12d$ digits long and so on. In general, if the coefficients of P_i are u_i digits long, then approximately $u_{i+2} = 2u_{i+1} + u_i$. Hence, approximately, $u_i = (1 + \sqrt{2})^{i-1} d$, $1 + \sqrt{2}$ being the dominant root of $x^2 = 2x + 1$. Thus, for the Euclidean algorithm, the lengths of the coefficients increases exponentially and hence so does the computing time, a similar but more complex analysis applying for multivariate polynomials. We estimate that the computing time in Table 1 for applying the Euclidean algorithm to the univariate polynomials of degree 15 would be of the order of one week, and would produce integers about one million decimal digits long!

For the other three algorithms the situation is entirely different. Let P_1 and P_2 be univariate polynomials of degree n with integer coefficients not exceeding d decimal digits. Let $P_1, P_2, S_3, \cdots, S_k$ be a subresultant p.r.s. for P_1 and P_2. Each coefficient of S_k is the determinant of a matrix of order $2(n - n_{k-1} + 1)$, each element of which is a coefficient of P_1 or P_2. By Hadamard's Theorem [5, pp. 78–79], the absolute value of this determinant does not exceed $10^{2d(n-n_{k-1}+1)} \cdot (2(n - n_{k-1} + 1))^{n-n_{k-1}+1}$. Thus the coefficients of S_k are at most $(n - n_{k-1} + 1) \cdot (2d + \log_{10} 2(n - \bar{n}_{k-1} + 1))$ decimal digits long. If $P_1, P_2, \cdots, P_k$ is the corresponding primitive p.r.s., the same bound applies to the coefficients of P_k, since $S_k = a_kP_k$ for some integer a_k. As already noted, both the ALPAK algorithm and the primitive p.r.s. algorithm compute the primitive p.r.s., but in different ways.

TABLE 1. Computing Times in Minutes

Degree	*Euclidean algorithm*	*ALPAK algorithm*	*Primitive p.r.s. algorithm*	*Reduced p.r.s. algorithm*
		Univariate Polynomials		
5	.0034	.018	.009	.0043
10	.94	.15	.064	.023
15		.51	.22	.077
20		1.19	.51	.21
25		2.29	1.06	.43
30		3.81	1.79	.78
35			3.25	1.48
		Bivariate Polynomials		
3		.30	.02	.015
4		7.03	.35	.062
5			4.29	.23
6				.67
7				1.81

The Alpak algorithm computes more g.c.d.'s of coefficients, presumably in an attempt to reduce the size of coefficients of intermediate polynomials computed between successive terms of the primitive p.r.s. The experiments reported in Table 1 strongly indicate that this extra effort falls far short of being adequately compensated, particularly for multivariate polynomials.

When P_1 and P_2 have a normal p.r.s., these same coefficient bounds apply to the reduced p.r.s. since then, by Corollary 1.3, the reduced p.r.s. and subresultant p.r.s. agree except for signs. For a nonnormal p.r.s. these coefficient bounds do not apply, and at present we have no theory to indicate that the reduced p.r.s. algorithm would still be more efficient than the primitive p.r.s. algorithm. We have, however, accumulated considerable experimental evidence that, in practice, deviations from normality are both rare and small (say in the sense that the expectation of $\sum_{i=2}^{k-2} \delta_{i-1}(\delta_i - 1)$ is small). For this reason, we have not at this time programmed the algorithm provided by Theorem 2 for computing a subresultant p.r.s.

While there is reason to believe that nonnormal p.r.s.'s occur infrequently, the existence of nonnormal p.r.s.'s is an entirely different matter. In fact, if $n_1, n_2, \cdots, n_k$ is any sequence of integers satisfying $n_1 \geq n_2 > n_3 > \cdots > n_k \geq 0$, there is a p.r.s. $P_1, P_2, \cdots, P_k$ such that $\deg(P_i) = n_i$ for $1 \leq i \leq k$. For, let $\delta_i = n_i - n_{i+1}$ and let Q_i be any polynomial of degree $\delta_{i-1}(2 \leq i \leq k-1)$. Also, let P_{k-1} and P_k be any polynomials of degrees n_{k-1} and n_k, respectively. By induction on i, define $P_{k-i} = P_{k-i+1}Q_{k-i+1} + P_{k-i+2}$ (for $2 \leq i \leq k-1$). Assuming, by induction, that $\deg(P_{k-i+1}) = n_{k-i+1}$ and $\deg(P_{k-i+2}) = n_{k-i+2}$, we have $\deg(P_{k-i+1}Q_{k-i+1}) = \deg(P_{k-i+1}) + \deg(Q_{k-i+1}) = n_{k-i+1} + \delta_{k-i} = n_{k-i} > \deg(P_{k-i+2})$, and hence $\deg(P_{k-i}) = \deg(P_{k-i+1}Q_{k-i+1}) = n_{k-i}$. Hence $P_1, P_2, \cdots, P_k$ is a p.r.s. A p.r.s. constructed in this way is artificial, however, in the sense that the remainder equations $a_iP_i = P_{i+1}Q_{i+1} + b_iP_{i+2}$ are all satisfied with $a_i = b_i = 1$. We do not know how to construct a p.r.s. with arbitrarily prescribed n_i for which this is not so.

Another theoretical problem relating to p.r.s. calculation is the extent to which the subresultant p.r.s. can deviate from the corresponding primitive p.r.s., where P_1 and P_2 are primitive. In this connection, note that since $\mathfrak{L}(P_1)$ and $\mathfrak{L}(P_2)$ are the only nonzero elements of the first column of the Sylvester matrix of P_1 and P_2, any common divisor of $\mathfrak{L}(P_1)$ and $\mathfrak{L}(P_2)$ is a divisor of each subresultant of P_1 and P_2. Consideration of other columns leads to similar observations. Apart from this remark, this appears to be a question about which little is known. The available experimental evidence indicates, however, that the deviation is ordinarily so small that the reduced p.r.s. algorithm is faster.

Since each pair of polynomials represented by Table 1 is relatively prime, one may ask whether a similar comparison would result for pairs which are not relatively prime. The answer is no. The data so far collected is too scanty to include, but it indicates that as the g.c.d. increases in degree the primitive p.r.s. algorithm improves relative to the reduced p.r.s. algorithm and may even be slightly faster in extreme cases. This slight advantage in these cases would seem to be far from adequate, however, to compensate the primitive p.r.s. algorithm for its relative inefficiency in the other cases.

The section of Table 1 applying to univariate polynomials displays well the dependency of computing times on the degree n of the initial polynomials. It is possible to give an argument supporting the view that this dependency can be approximated by a polynomial in n of degree 4 (this applies to all except the Euclidean algorithm). Table 1 does not, however, indicate the dependency of computing time on d, the number of decimal digits in the coefficients of the initial polynomials (for fixed n). This dependency can be approximated by a quadratic polynomial in d. For example, application of the reduced p.r.s. algorithm with $n = 15$ resulted in computing times of .077, .18 and .53 for $d = 2$, 4 and 8, respectively.

Suppose one wishes to compute the g.c.d. of univariate polynomials, P_1 and P_2, with elements of R, the field of rational numbers, as coefficients. Since R is a field, there is essentially only one p.r.s. $P_1, P_2, \cdots, P_k$ with elements P_i in $\mathcal{P}(R)$. If one computes this p.r.s. in order to obtain the g.c.d. of P_1 and P_2, there are two cases according as one does or does not represent each rational number with relatively prime numerator and denominator. In the first case the number of g.c.d.'s of integers computed is far larger than in the use of the ALPAK algorithm, and the computing time is correspondingly large. In the second case, on the other hand, the integers clearly grow exponentially as in the Euclidean algorithm. It seems clear, therefore, that the recommended procedure is to replace P_1 and P_2 with primitive associates, $\bar{P}_1$ and $\bar{P}_2$, with integer coefficients and apply the reduced p.r.s. algorithm. A similar analysis applies to multivariate polynomials with rational coefficients.

Finally, a few remarks are added with regard to the reduced p.r.s. algorithm for computing the resultant R of polynomials P_1 and P_2 (with respect to the main variable). One computes the complete reduced p.r.s. $P_1, P_2, \cdots, P_k$. Then, by Theorem 1, part (a), $\pm R$ is obtained by dividing $P_k^{\delta_{k-1}}$ by $[\prod_{i=2}^{k-1} c_i^{\delta_{i-1}(\delta_i-1)}]$, since $P_k = c_k$. It is clear that the computing time is the same as in the reduced p.r.s. g.c.d. algorithm, except that in computing the resultant some additional time is required for the terminal division whenever the resultant is nonzero (P_1 and P_2 have no common factor of positive degree in the main variable) and the p.r.s. is nonnormal.

In this connection, however, it should be mentioned that Williams [6] proposes (translated into our terminology) computing the complete Euclidean p.r.s. $P_1, P_2, \cdots, P_k$ as a means of obtaining the "eliminant" of P_1 and P_2. It is not clear whether Williams realizes that this "eliminant" may differ essentially from the resultant. In any case it seems desirable to point out that his Theorem IV [6, p. 32] is false. Translated into our terminology, it states the following: Let $P_1(x_1, \cdots, x_m, y)$, $P_2(x_1, \cdots, x_m, y), \cdots, P_{k-1}(x_1, \cdots, x_m, y), P_k(x_1, \cdots, x_m)$ be a complete Euclidean p.r.s. with respect to the main variable y. Let $\alpha_1, \cdots, \alpha_m, \beta$ be complex numbers such that $P_k(\alpha_1, \cdots, \alpha_m) = P_{k-1}(\alpha_1, \cdots, \alpha_m, \beta) = 0$ and $P_2(\alpha_1, \cdots, \alpha_m, y) \neq 0$. Then $P_1(\alpha_1, \cdots, \alpha_m, \beta) = P_2(\alpha_1, \cdots, \alpha_m, \beta) = 0$. A simple counterexample is obtained by taking $m = 1$, $P_1(x, y) = xy^2 + 2y + 1$, $P_2 = xy^2 + y + x$, $\alpha = 0$ and $\beta = 1$. Then $P_3(x, y) = xy + x - x^2$, $k = 4$ and $P_4(x) = x^5 - 2x^4 + 3x^3 - x^2$. The erroneous proof apparently assumes that $P_1(\alpha_1, \cdots, \alpha_m, y), P_2(\alpha_1, \cdots, \alpha_m, y), \cdots, P_k(\alpha_1, \cdots, \alpha_m)$ is a p.r.s., an assumption which can be realized if we add the hypotheses $A_i(\alpha_1, \cdots, \alpha_m) \neq 0$, for $2 \leq i \leq k - 1$, where $A_i(x_1, \cdots, x_m)$ is the leading coefficient of $P_i(x_1, \cdots, x_m, y)$. This error, together with the impracticality of computing a Euclidean p.r.s., vitiates much of the content of [6].

ACKNOWLEDGMENT. I would like to thank Professor Abraham Robinson, since his remarks provided the original stimulus for this investigation.

RECEIVED MAY, 1966

REFERENCES

1. Collins, G. E. Polynomial remainder sequences and determinants. *Amer. Math. Mon. 73*, 7 (Aug.-Sept. 1966), 708-712.
2. Brown, W. S., Hyde, J. P., and Tague, B. A. The ALPAK system for non-numerical algebra on a digital computer. *Pt. I: Bell System Tech. J. 42*, 5 (Sept. 1963), 2081-2119. *Pt. II: Ibid. 43*, 2 (March 1964), 785-804. *Pt. III: Ibid. 43*, 4 (July 1964), 1547-1562.
3. Bocher, M. *Introduction to Higher Algebra.* MacMillan Co., New York, 1907.
4. Collins, G. E. PM, a system for polynomial manipulation. *Comm. ACM 9*, 8 (Aug. 1966), 578-589.
5. Bodewig, E. *Matrix Calculus* (2nd Ed.). North-Holland Publishing Co., Amsterdam, 1959.
6. Williams, Leland H. Algebra of polynomials in several variables for a digital computer. *J. ACM 9*, 1 (Jan. 1962), 29-40.