J. Symbolic Computation (1987) **3**, 3–38

History and Basic Features of the Critical-Pair/Completion Procedure

BRUNO BUCHBERGER

Johannes-Kepler-University, Department of Mathematics, A4040 Linz, Austria

A synopsis of the critical-pair/completion approach for solving algorithmic problems in theorem proving, finitely generated algebras and rewrite systems is given. The emphasis is on tracing the main intuition behind the approach, which consists in considering "critical pairs". These are the "first possible" situations where "superpositions" may occur. Extensive references to the original literature are provided. Some directions for future research are outlined. The presentation is biased towards the development of the approach in algorithmic polynomial ideal theory.

Introduction

The critical-pair/completion (CPC) approach is an algorithmic problem solving strategy that combines two *key ideas*:

completion and
the formation of critical pairs.

The CPC technique was independently initiated by *three papers in the mid sixties* in three seemingly separate areas:

automated theorem proving
polynomial ideal theory and
solution of word problems in universal algebras.

In retrospect, however, it turns out that the key ideas of the CPC approach were implicitly already around in the early *forerunners*.

In the 20 years since 1965 the CPC approach has found more and more useful *applications* in various areas of algorithmic problem solving. In the most recent years research on applications of the CPC technique has been particularly intensive and successful.

Various technical questions for improving and analysing CPC algorithms and for broadening the scope of applicability of the CPC approach have been and are studied by an increasing research community: *termination*, *strategies* for selecting critical pairs, *criteria* for omitting certain critical pairs, *complexity* of CPC computations. Various *implementations* of CPC algorithms have been described in the literature, starting from the early implementations in the mid sixties and proceeding to sophisticated implementations as parts of software systems for symbolic computation.

The *main goal of this paper* is to elaborate the basic ideas and major advances in CPC research by pointing to some key papers. We hope this will contribute to a further cross-

0747–7171/87/010003+36 $03.00/0

fertilisation between the different areas involved in CPC research and applications. Elaborating the basic ideas of the class of CPC algorithms may serve also as a case study for the more general objective of establishing certain basic *algorithm types* in computer science as a natural analogue to the concept of *data types.*

I apologise that my presentation is biased towards polynomial ideal theory both in the application section and in the technical section. This is certainly due to my own involvement in this root of CPC research and also due to my lack of expertise in the other branches. However, this paper addresses the participants of a rewrite rules conference. I therefore hope that polynomial ideal theory is that branch of CPC research that may provide some information supplementary to the ordinary background of researchers working in rewrite rule techniques and, hence, may be of some interest to the audience of this conference.

Most probably the readers will not be satisfied with my tracing back the historical roots of the CPC approach either: only during writing this survey I detected how difficult it is to give fair and complete credit to the work of all the people who have been involved in CPC research. Fortunately, there is a "bottom element" to the historical priority graph in CPC research: it is well known that Euclid's algorithm for polynomials is (an instance of a) CPC algorithm. Recently, as a curiosity, it has been shown also that Euclid's algorithm for integers may be viewed as an instance of CPC algorithms, (see Loos, 1981; Buchberger, 1983). Thus, Euclid spares me the trouble of tracing priorities too pedantically.

Key Ideas and Basic Structure of the CPC Approach

Typically, the CPC approach can be applied when one has:

a *set* T of (linguistic, algebraic, symbolic) objects
together with a binary "*reduction*" relation $\rightarrow$ on T that is generated by a set F of (finitely) many "*patterns*" F
and one wants to solve "*word problems*" of the kind
"for $s, t \in T$:
is (s, t) in the reflexive, symmetric, transitive closure of $\rightarrow$?"
or problems that can be reduced to such problems.

SETS OF OBJECTS

Typical sets of objects are:

the set of clauses over an alphabet of function and predicate symbols,
the set of polynomials over a coefficient ring,
the set of equations between terms over an alphabet of function symbols,
the set of words over a finite alphabet of constants.

REDUCTION RELATIONS GENERATED BY PATTERNS

For generating "*reduction*" relations $\rightarrow$ from patterns one starts from a set F of finitely many "*patterns*" ("*basic reductions*" or "*rules*") $(s, t) \in T \times T$ of reductions that, by definition, are assumed to be in $\rightarrow$, i.e. one stipulates:

$$\text{for all } s, t \in T: \quad \text{if } (s, t) \in F \quad \text{then } s \rightarrow t. \qquad (\textit{pattern rule})$$

(We write $s \to t$ instead of $(s, t) \in \to$. For s and t in a pattern $(s, t) \in F$ we sometimes say "*left-hand side*" and "*right-hand side of the pattern*" respectively.)

In addition, one has an (infinite) set Σ of "*multipliers*" ("substitutions", "operators") that can be applied to objects of T and yield objects in T. (We write σs for the application of the substitution σ to the object s). Now, for the application of substitutions to both sides of patterns one stipulates:

$$\text{for all } s, t \in T, \sigma \in \Sigma\text{: if } (s, t) \in F \text{ then } \sigma s \to \sigma t. \qquad \textit{(multiplier rule)}$$

Thus, by applying all substitutions σ, each of the patterns $s \to t$ in F generates a whole "*spectrum*" $\{\sigma s \to \sigma t : \sigma \in \Sigma\}$ of reductions.

Finally, in the cases amenable to the CPC approach, one has a concept of "*places*" in the objects of T and, correspondingly, a concept of "*replacement*" in objects. Let P be the set of possible places in objects, let us write r/u for "the object located at place u in the object r" and let us write $r[u \to t]$ for "the object resulting from replacing the subobject located at place u in r by the object t". Then one stipulates

$$\text{for all } r, s, t \in T, \sigma \in \Sigma, u \in P\text{:}$$

$$\text{if } (s, t) \in F \text{ and } r/u = \sigma s \text{ then } r \to r[u \leftarrow \sigma t]. \qquad \textit{(replacement rule)}$$

(This means that objects r can be reduced by replacing a subobject "of the form σs" by σt whenever $s \to t$ is a pattern in F.)

The relation $\to$ generated from a set F of patterns in the way described above, i.e. by applying the pattern rule, substitution rule and replacement rule, will be called "the *reduction relation generated by F*" (denoted by $\to_F$). This notion is not meant to be a serious mathematical "definition" on which the rest of the paper could be based in a closed deductive presentation. The description given here, rather, is an attempt to extract as many common features as possible from the various situations in which CPC algorithms have been applied successfully. An axiomatic approach to CPC algorithms exclusively based on the above three notions of patterns, multipliers and replacements seems to be promising. However, this has not yet been achieved satisfactorily although much progress has been made in general formulations of the CPC approach (see the section on unifying approaches).

WORD PROBLEMS

In situations where one has a reduction relation $\to_F$ generated by patterns, typically, many algorithmic problems can be reduced to the following problem (the *uniform word problem for reductions relations generated by patterns in T*):

for arbitrary $s, t \in T$ and (finite) $F \subset T \times T$ decide whether $s \leftrightarrow_F^* t$.
($\leftrightarrow^*$ denotes the reflexive, symmetric, transitive closure of the binary relation $\to$.)

Examples of such problems are:

the problem of deciding whether the empty clause can be derived from a set of clauses,
the problem of constructing a vector space basis for the residue ring modulo a multivariate polynomial ideal,
the problem of deciding whether a given equation can be derived from the axioms of a given equational theory,
the reachability problem for reversible Petri nets,

the problem of constructing direct implementations for abstract data types,
the problem of finding a generating set for the module of all syzygies of a polynomial ideal,
the problem of solving algebraic systems of equations,
and many others (see the section on applications).

FINITE TERMINATION

So far we have not yet discussed why we spoke about "reductions" when we introduced the concept of relations generated by patterns. Actually, in algorithmic problem solving one is only interested in those situations where by one "step" $s \rightarrow_F t$ the "complexity" or "size" of s is "reduced". Whatever the notion of complexity is, one of course would not like to admit that the complexity can be reduced infinitely often. This means that one normally is interested only in *noetherian* relations $\rightarrow_F$. (A binary relation $\rightarrow$ on T is called noetherian iff there is no infinite chain $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \ldots$. Sometimes these relations are also said to have the finite termination property.) Thus, normally, when speaking about a reduction relation generated by F we presuppose that the relation, in addition to being generated by the process described above, is noetherian. On the other hand, it may sometimes be reasonable to disregard the question of finite termination and still try to apply the CPC approach.

SOLVING WORD PROBLEMS: CHURCH–ROSSER PROPERTY AND CONFLUENCE

Given a (reduction) relation $\rightarrow$ on T, it is clear that if s and t have "a common successor" then $s \leftrightarrow^* t$. In general, the converse does not hold. Noetherian relations $\rightarrow$ for which the converse does hold have a decidable word problem. We state these well known and easy facts more formally (for proof details see, for example, Buchberger & Loos, 1982, p. 27).

Let $\rightarrow$ be a binary relation on T. As usual, the *inverse* and the *reflexive-transitive closure* of $\rightarrow$ are denoted by $\leftarrow$ and $\rightarrow^*$ respectively. Furthermore, for $s, t \in T$:

$s \downarrow t$ (s and t have a *common successor*): $\Leftrightarrow (\exists\, r)(s \rightarrow^* r \leftarrow^* t)$,
$\underline{s}$ (s is in *normal form*): $\Leftrightarrow$ not $(\exists\, t)(s \rightarrow t)$.

DEFINITION: $\rightarrow$ has the *Church–Rosser property* iff $(\forall\, s, t)(s \leftrightarrow^* t \Rightarrow s \downarrow t)$.

LEMMA (*Decidability of Church–Rosser relations*): *If* $\rightarrow$ *is noetherian and has the Church–Rosser property then* $\leftrightarrow^*$ *is decidable.*

PROOF (Sketch): The following function S is a canonical simplifier for $\leftrightarrow^*$:

$$S(s) := \textit{if } s \text{ is in normal form } \textit{then } s \textit{ else } S(Sel(s)),$$

where *Sel* is a "selector" function for $\rightarrow$. Hence, we have the following decision algorithm:

$$s \leftrightarrow^* t \quad \text{iff } S(s) = S(t).$$

(A function S defined by a recursion of the above type is called a *normal form algorithm.*

A computable function $Sel: T \to T$ is called a *selector function* for $\to$ iff for all $s \in T$ that are not in normal form: $s \to Sel(s)$.

A computable function $S: T \to T$ is called a *canonical simplifier* for an equivalence relation $\sim$ on T iff for all $s, t \in T$:

$$S(s) \sim s, \qquad \text{(closure)}$$

$$\text{if } s \sim t \quad \text{then } S(s) = S(t) \qquad \text{(uniqueness).}$$

Note that in the above proof we need the existence of a computable selector function and the decidability of the predicate "is in normal form". We do not explicitly mention these assumptions in the lemma in order not to distract the attention from the crucial points. Actually, in practically interesting cases the validity of these two assumptions is no problem.)

The proof tells us that, in the case of noetherian Church–Rosser relations $\to$, for deciding $s \leftrightarrow^* t$ we only need to reduce s and t iteratively by "applying" $\to$ until we arrive at normal forms s' and t'. Then $s \leftrightarrow^* t$ iff $s' = t'$.

Of course, in general, it is not easily possible to determine whether a given relation $\to$ has the Church–Rosser property since the condition in the definiens in general involves tests for infinitely many pairs (s, t) each of which may give rise to infinitely many attempts to find common successors for s and t. The following lemma gives an equivalent formulation that presents an "easier" but still non-constructive test.

DEFINITION: $\to$ is *confluent* iff $(\forall r, t, s \in T)(s \leftarrow^* r \to^* t \Rightarrow s \downarrow t)$.

LEMMA (*Reduction of the Church–Rosser property to confluence*): *$\to$ has the Church–Rosser property iff $\to$ is confluent.*

LEMMA (*Alternative formulation of confluence*): *$\to$ is confluent iff* $(\forall r, s, t \in T)(\underline{s} \leftarrow^* r \to^* \underline{t} \Rightarrow s = t)$.

SOLVING THE WORD PROBLEM: THE IDEA OF COMPLETION

Stated in the context of confluence the idea of *completion* is straightforward:

—Given a set F of patterns we try to find a set G such that
$\leftrightarrow_F^* = \leftrightarrow_G^*$ and
$\to_G$ has the Church–Rosser property.
(By the lemma on the decidability of Church–Rosser relations, we then have a decision algorithm for $\leftrightarrow_F^*$.)

(A set of patterns having the property that $\to_G$ has the Church–Rosser property will be called a *completed set*.)

—The lemma on the reduction of the Church–Rosser property to confluence (using the alternative formulation for confluence) suggests the following procedure for finding a suitable set G:

"*Algorithm*" (Completion of a set of patterns):

$G := F$

$B := \{(s, t) : (\exists r)\ (s \leftarrow_F^* r \rightarrow_F^* t)\}$.

while $B = 0$ *do*

$(s, t) :=$ an element in B
$B := B - \{(s, t)\}$

if $s \neq t$ *then*
analyse (s, t)
$G := G \cup \{(s, t)\}$.

The completion proceeds by locating all situations (r, s, t) in which confluence is injured. In these situations, as a remedy, $s \rightarrow t$ (or $t \rightarrow s$) is adjoined to the set of "patterns" (dependent on the analysis whether adjoining $s \rightarrow t$ or $t \rightarrow s$ leaves the finite termination property untouched. If both possibilities destroy the finite termination property then this procedure must be terminated "with failure".) It should be clear that adjoining these $s \rightarrow t$ as patterns preserves the condition $\leftrightarrow_F^* = \leftrightarrow_G^*$ and, if the algorithm terminated, $\rightarrow_G$ would have the confluence (and, hence, the Church–Rosser) property (G has been "*completed*"). However, in general, B is an infinite set and, hence, the above construction is not algorithmic.

The above completion procedure can be slightly improved by using Newman's lemma on *local confluence*.

DEFINITION: $\rightarrow$ is *locally confluent* iff $(\forall r, s, t \in T)(s \leftarrow r \rightarrow t \Rightarrow s \downarrow t)$.

LEMMA (*Reduction of confluence to local confluence; (Newman, 1942)): Let* $\rightarrow$ *be noetherian. Then* $\rightarrow$ *is confluent iff* $\rightarrow$ *is locally confluent.*

PROOF: By noetherian induction. The lemma is due to Newman (1942). In is full generality it has been proven by Huet (1977). The proof may also be found, for example, in Buchberger & Loos (1982).

LEMMA (*Alternative formulation of Newman's lemma): Let* $\rightarrow$ *be noetherian and S be a normal form algorithm for* $\rightarrow$. *Then*

$$\rightarrow \text{ is confluent iff } (\forall r, s, t \in T)(s \leftarrow r \rightarrow t \Rightarrow S(s) = S(t)).$$

Using the alternative formulation of Newman's lemma it is clear that in the above completion procedure the second statement can be replaced by

$$B := \{(s, t) : (\exists r, s', t')(s' \leftarrow_F r \rightarrow_F t',\ s = S(s'),\ t = S(t'))\}.$$

Intuitively, using this B, "fewer" situations have to be checked than in the first formulation of the completion procedure. However, in general, B is still an infinite set.

A further improvement of the completion construction is suggested by taking into account that the reduction relations $\rightarrow_F$ "generated by patterns F" are built by substitutions and then replacements. Intuitively, one may expect that a sound notion of replacement has the following *compatibility* property with respect to reduction:

$$(\forall r, s, t \in T, u \in P)(s \rightarrow t \Rightarrow r[u \leftarrow s] \rightarrow r[u \leftarrow t]). \qquad (\textit{compatibility})$$

(In fact, in some important cases, compatibility in this strong form is not available. This raises technical difficulties!) In case compatibility holds it is clear that it suffices to consider confluence on the "spectra" of the patterns rather than general confluence, i.e. in the completion procedure the second statement can be replaced by

$$B := \{(s, t) : (\exists\, r \in \text{spectrum } (F), s', t')(s' \leftarrow_F r \rightarrow_F t', s = S(s'), t = S(t'))\},$$

where spectrum $(F) := \{\sigma s : (\exists t)((s, t) \in F), \sigma \in \Sigma\}$.

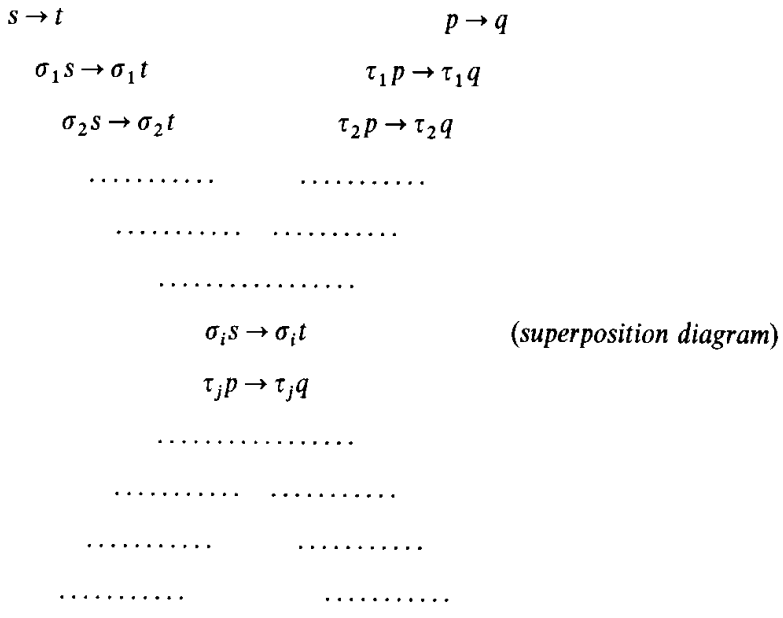
Again, this definition of B is a step towards turning the completion procedure into a real algorithm. It even gives us some hint how to exhaust B by generating finite subsets of B: in one step one could consider those r in the spectrum (F) that are generated by a fixed substitution σ from the left-hand sides of patterns in F. (The expert reader will note that in this introduction we oversimplify the situation for the sake of bringing to light the key ideas at the cost of details: actually one often has to consider $r \in$ spectrum (F) that are subobjects in other $r' \in$ spectrum (F).)

SOLVING THE WORD PROBLEM: THE IDEA OF CRITICAL PAIRS

The analysis given so far shows us that for completing F (and, hence, solving the word problem for $\rightarrow_F$) we should look at the "spectra" of the patterns $s \rightarrow t$ in F

$s \rightarrow t$
$\sigma_1 s \rightarrow \sigma_1 t$
$\sigma_2 s \rightarrow \sigma_2 t$
. . . . . .
. . . . . .
. . . . . .

and locate objects r that can be conceived as the left-hand sides $\sigma_i s = \tau_j p$ of reductions in two different spectra:



(superposition diagram)

In such situations we have for $r := \sigma_i s = \tau_j p$

$$\sigma_i t \leftarrow r \rightarrow t_j q$$

and we only need to check whether $S(\sigma_i t) = S(\tau_j q)$. From the suggestive superposition diagram above the *idea of critical pairs* may come naturally to ones mind:

> Instead of considering the (infinitely) many objects r that can be conceived as the left-hand sides of reductions in two different spectra, does it suffice to consider the "*first possible*" situations in which a "*superposition*" happens and to remove a possible injury of local confluence in these situations? These situations are called the "critical situations" and the pairs $(\sigma_i t, \tau_j q)$ are called the "*critical pairs*".
>
> Differently stated: the idea of critical pairs is the desire to "come as quickly as possible to a situation where something interesting can happen by the interaction (interference, superposition) of two patterns".

In fact, it turned out that this idea works in many interesting examples. From the superposition diagram it should be clear that, as minimal requirements, for the idea to work one needs

> a notion of "*the first possible superposition situation*" for two patterns, i.e. a superposition situation from which all other superposition situations for these two patterns can be generated by application of multipliers (substitutions).

More formally (but, again, not meant as a "definition"): given two "patterns" $s \rightarrow t$ and $p \rightarrow q$ the two objects x, y form a *critical pair* for the two patterns if

> there exist two substitutions σ and τ such that
> $\sigma s = \tau p$
> for all substitutions σ', τ' for which $\sigma' s = \tau' p$
> there exists a substitution χ such that $\sigma' s = \chi \sigma s$,
> and $x = \sigma t$, $y = \tau q$
> (i.e. x and y result from applying the patterns $s \rightarrow t$ and $p \rightarrow q$ to a "first possible superposition" of s and p.)

In more general situations, where we consider objects σs that are subobjects of τp or where the set of all superposition situations of two patterns are generated by more than one generating situation, more than one critical pair can correspond to two patterns.

Combining the idea of completion with the idea of critical pairs, finally, leads to the following CPC algorithm schema.

Algorithm (Structure of critical-pair/completion algorithms):

$G := F$

$B := \bigcup_{f,g \in G}$ set of critical pairs of f and g

while $B \neq \emptyset$ *do*

$(s, t) :=$ an element in B
$B := B - \{(s, t)\}$
$(s, t) := (S(s), S(t))$

if $s \neq t$ *then*

analyse (s, t)

$B := B \cup \bigcup_{(p,q) \in G}$ set of critical pairs of (s, t) and (p, q)

$G := G \cup \{(s, t)\}$.

Introducing critical pairs was the crucial advance in the development of the CPC approach because in many typical application areas the sets of critical pairs (for the finitely many $f, g \in G$) are finite and, hence, the completion construction has a chance to become a real algorithm. Still, the termination of CPC algorithms may remain a difficult problem for three reasons. First, in the particular context where one wants to apply the CPC approach, it may be difficult to establish a Noetherian ordering on the objects that is compatible with reduction. Second, when analysing (s, t) it may turn out that the finite termination property can not be guaranteed after augmenting G by (s, t). And third, in the while-loop, B will be diminished and increased alternately and it is by no means trivial that it ever becomes empty.

Note that the above structure of CPC algorithms does not reflect any of the more subtle details of the approach, for example the strategy of keeping patterns in G reduced with respect to all other patterns in G; see the section on strategies.

Three CPC Algorithms in the Mid Sixties

The CPC approach was introduced in three papers in the mid sixties in three different areas that, at first sight, seem to be far apart:

universal theorem proving
polynomial ideal theory
word problems in universal algebras.

These three papers contained complete correctness proofs for the respective algorithms and described computer implementations of the algorithms. We give a short review of these three algorithms.

THE RESOLUTION PROCEDURE (Robinson, 1963, 1965)

The original problem

Given F, a set of clauses in first-order predicate logic.
Semi-decide whether F is unsatisfiable.

Robinson's resolution algorithm needs some distortion in order that it can be viewed as a CPC algorithm. However, it surely shows the two key ideas of the CPC approach: completion and critical pairs.

Objects: The set T of clauses.

Patterns: The clauses in F. Each clause $\{L_1, \ldots, L_i, M_1, \ldots, M_j\}$ can be viewed in several ways as a "pattern" (s, t) depending on which of the positive literals $L_1, \ldots, L_i$ or negative literals $M_1, \ldots, M_j$ is taken as the "left-hand side" s. Note that, in the context of

the resolution algorithm, clauses are the objects *and* the rules. We will have a similar situation also in the context of polynomials (and general rings, see the section on generalisations.)

Multipliers: The substitutions σ of first-order predicate logic can be viewed as the multipliers. Each clause $C = (s, t)$ generates a whole "spectrum" of clauses σC.

Replacements: A special case of resolution may be viewed as a replacement: $C \to C[u \leftarrow \sigma t]$, if the literal L at "place" u is the negation of σs, where (s, t) is a clause in F, conceived as a "pattern".

Critical pairs: By formal distortion, one basic step in the resolution algorithm can be viewed as forming a critical pair: for two clauses (s, t) and $(-p, q)$ in F (where the minus sign stands for "not"), conceived as patterns, "*most general unifiers*" σ and τ are determined such that $\sigma s = \tau p$. From σs, in one reduction, one obtains σt and $-\tau q$. $(\sigma t, -\tau q)$ could be added to F now. (No simplification by a "normal form algorithm" S is foreseen in the resolution procedure!) Instead, $(\sigma t + \tau q)$ is added to F, where the plus sign stands for "or". $((\sigma t, -\tau q)$, formally, would not be a clause!). The notion of a most general unifier and the *unification algorithm* that determines the most general unifier for two expressions (if it exists), introduced in (Robinson, 1965) is an important concept that has motivated a whole stream of research in symbolic computation.

Completion: Adding $(\sigma t + \tau q)$ to F is the completion step in the resolution procedure.

Completed sets: They do not play an explicit role in the context of resolution because in the successful cases the procedure stops when the empty clause is generated and the set of clauses G, generated until then, is not used further.

Remarks: It is interesting to note that, disregarding formal details, the key idea of critical pairs seems to have been very clear in the intention of J. A. Robinson because (Robinson, 1979, p. 292) he writes: "The idea that, instead of *trying* all instantiations over the Herbrand Universe, one might *predict* which ones would produce a 'winning combination' by using what we have called the Unification Algorithm, ...". In a colloquium lecture at ETH (Zürich, 1978) I proposed to consider the resolution procedure as a CPC algorithm using the sketch given above. Meanwhile the concept of the resolution procedure as a CPC algorithm has been worked out much more specifically by J. Hsiang and others based on the Peterson–Stickel version of the Knuth–Bendix algorithm, see the section on unifying approaches. We still think it would be worthwhile to look for a unifying approach that is based on an axiomatisation of "multipliers" and "replacement". Also including "simplification by resolution" in the resolution procedure seems to be promising (for simplification, only matching, not unification, is necessary; see also the concept of "narrowing" in a later section). Of course, it should also be mentioned that some details of the resolution procedure in its original form are far apart from the general structure of CPC algorithms sketched in the preceding section. For example, termination of the reduction process, orientation of rules, the Church–Rosser property and local confluence do not play any explicit role in the original resolution context. Also, at first sight, the problem solved by the resolution procedure is not a word problem. (However, note that F is insatisfiable iff "the empty clause is reachable from F by resolution steps").

Introductory reading: (Robinson, 1979) is the authoritative presentation of the resolution method.

AN ALGORITHM FOR POLYNOMIAL IDEALS (Buchberger, 1965)

The original problem

Given F, a finite set of multivariate polynomials over a field K.
Find a linearly independent basis A and the multiplication table M for the associative algebra $K[x_1, \ldots, x_n]/\text{Ideal}(F)$, where $\text{Ideal}(F)$ is the ideal generated by F.

The algorithm presented in (Buchberger, 1965) has all the structural characteristics of a CPC algorithm. Using this algorithm the above problem can be solved as follows:

1. Construct a completed set G using the algorithm.
2. A: = the set of (the residue classes of) all power products that are not multiples of leading power products of polynomials in G.
3. The linear representation of the products $u \,.\, v$ $(u, v \in B)$ can be obtained by reducing $u \,.\, v$ to normal form using $\rightarrow_G$.

Objects: T: $= K[x_1, \ldots, x_n]$, the set of polynomials over the field K.

Patterns: The polynomials in F. In order to conceive a polynomial f as a "pattern" we present it in the form $f = s - t$, where s is the "leading power product" in f with respect to a given linear ordering of the power products and t comprises the remaining monomials in f. (Without loss of generality we may assume that the coefficient of this power product is 1. In the original paper we always took the "lexicographical ordering graded by degrees" as a fixed ordering for the power products. Later it turned out that the algorithm can be carried through for a whole class of orderings that can be characterised by two easy and natural axioms.)

Multipliers: The monomials (coefficients times power products) serve as "multipliers". The "application" σs of a multiplier σ to a polynomial s is just the product of σ and s.

Replacements: The "places" of a polynomial are the power products. $r[u \leftarrow t]$ must be interpreted as "the polynomial that results from replacing the monomial at place u in r by the polynomial t". r/u must be interpreted as "the monomial at place u in r". The general reductions generated by a set F of patterns, then, are reductions of the following form:

$$r \rightarrow r[u \leftarrow \sigma t], \quad \text{if } (s, t) \in F \quad \text{and } r/u = \sigma s.$$

(A careful definition of the notion of a "polynomial" is crucial in this context. The concept of a polynomial must not be confused with the concept of an arithmetical term. For an exact definition see, for example (Buchberger & Loos, 1982, p. 16). As one of the isomorphic models of the set of polynomials $K[x_1, \ldots, x_n]$ one can take the set $\{s : s \text{ is a function from } N_0{}^n \text{ in } K \text{ such that } s \text{ is zero for almost all arguments}\}$. In this model the "places", then, are just the tupels u in $N_0{}^n$. We wish to emphasise these well known distinctions because they sometimes still cause confusion and wrong "proofs" in papers on CPC algorithms and simplification modulo ideals in polynomial rings.)

Critical pairs: With the above interpretation of "application of multipliers" and "replacement" the definition of a critical pair can now be literally taken from the section

on the key ideas: Given (s, t) and $(p, q) \in F$ the two polynomials x, y form a *critical pair* for the two patterns if

there exist two multipliers σ and τ such that
$\sigma s = \tau p$
for all multipliers σ', τ' for which $\sigma' s = \tau' p$
there exists a multiplier χ such that $\sigma' s = \chi \sigma s$
and $x = \sigma t$, $y = \tau q$.

More explicitly, in order to find σ and τ, we only have to compute the *least common multiple* m of s and p. σ and τ, then, are just the monomials that satisfy $\sigma s = m$ and $\tau t = m$.

Completion: With the interpretations described above, the algorithm developed in (Buchberger, 1965) introduced exactly the structure of the CPC algorithm schema shown in the section on the key ideas. In the context of polynomials, adjoining (s, t) to G must be realised by adjoining $r := s - t$ to G. As a "pattern", r then splits into its leading power product and the remaining monomials. The step "analyse" is not necessary in the context of polynomials because $\rightarrow_G$ is noetherian for every G. The termination of the algorithm in the general case has been shown in (Buchberger, 1970) by rediscovering Dickson's lemma (Dickson, 1913). The original correctness proof of the algorithm was inductive in nature but not based on Newman's lemma. It can be based on Newman's lemma, see (Bachmair & Buchberger, 1980) for absorbing the set theoretical part of the underlying induction. However, the crucial part of the proof, which consists of exploiting the power of the critical pair construction, goes beyond Newman's lemma and must also cope with certain technical difficulties concerning the reduction process for polynomials.

Completed sets: The notion of a set G completed by the CPC algorithm was explicitly introduced in (Buchberger, 1966) by stating its ideal theoretically characteristic property: all polynomials in G can be reduced to zero by iteratively applying the reduction $\rightarrow_G$. Later, in (Buchberger, 1976), we called these sets *Gröbner bases* (for historical reasons, see the section on early forerunners.)

Remarks: We derived the crucial intuition behind our 1965 algorithm from drawing pictures showing the "spectra" of the leading power products of the polynomials in F in the way shown in the section on the key ideas and analysing the "first points where something interesting can happen". These points are the least common multiples of the leading power products. It is interesting to compare this with the intuition described by Robinson for his resolution procedure. Instead of reducing both polynomials x and y in a critical pair to normal form, in our algorithm, we reduce the difference $x - y$. If $h := S(x - y)$ is not equal to zero then h is adjoined to G. Computationally, this is slightly better because we need only one reduction to normal form instead of two. Logically, in the context of polynomials over K, these two procedures are equivalent. In (Buchberger, 1965) the difference $x - y$ of the components x and y of the critical pair corresponding to the polynomials p and q is called the "S-polynomial" of s and p.

Introductory reading: (Buchberger, 1985) gives an easy introduction to the algorithm for computing Gröbner bases and its many applications.

AN ALGORITHM FOR WORD PROBLEMS IN UNIVERSAL ALGEBRAS
(Knuth & Bendix, 1967)

The original problem (the word problem in universal algebras):

Given F, a finite set of identities described by pairs of first-order terms and two terms s and t.
Decide whether the identity "$s = t$" can be derived from the identities in F.

Again, the algorithm presented in (Knuth & Bendix, 1967) for the solution of this problem has all the structural characteristics of a CPC algorithm. Using this algorithm the above problem can be solved as follows:

1. Construct a completed set G using the algorithm.
2. (If the algorithm terminated successfully:)
   "$s = t$" is derivable from F (iff "$s = t$" derivable from G) iff $S(s)$ is identical to $S(t)$, where S is a normal form algorithm for the reduction $\rightarrow_G$.

Objects: T: = the set of first-order terms over a given alphabet of function symbols.

Patterns: Pairs (s, t) of terms ("identities"), where $s > t$ in some noetherian ordering $>$ of the terms. Term orderings suitable in this context must satisfy certain basic properties. (Essentially, if a term p reduces to a term q, then q must be smaller than p in the ordering.)

Multipliers: As in the resolution algorithm, the substitutions of terms for variables serve as the "multipliers".

Replacements: The "places" in a term are the places where the subterms occur. $r[u \leftarrow t]$ must be interpreted as "the term that results from replacing the subterm at place u in r by the term t". r/u must be interpreted as "the term at place u in r". The general reductions generated by a set F of patterns, then, are reductions of the following form:

$$r \rightarrow r[u \leftarrow \sigma t], \quad \text{if } (s, t) \in F \quad \text{and} \quad r/u = \sigma s.$$

Critical pairs: With the above interpretation of "application of multipliers" and "replacement" the definition of a critical pair could now be taken literally from the section on the key ideas. However, in the context of first-order terms, one has to consider also "superpositions" between terms and subterms of other terms and not only between terms and other terms. Given two patterns (s, t) and $(p, q) \in F$, the two terms x, y form a *critical pair* for the two patterns if

there exist two substitutions σ and τ and a place u such that
  $\sigma(s/u) = \tau p$,
  for all substitutions σ', τ' for which $\sigma'(s/u) = \tau' p$
    there exists a substitution χ *such that* $\sigma' = \chi\sigma$
  (and some technical conditions on variables hold),
and $x = \sigma t$, $y = \sigma s[u \leftarrow \tau q]$.

The substitutions σ and τ form a most general unifier exactly in the same sense as in the resolution algorithm. Actually, the same unification algorithm as in (Robinson, 1965) was also proposed in (Knuth & Bendix, 1967).

Completion: With the interpretations described above, the algorithm introduced in (Knuth & Bendix, 1967) has exactly the structure of the CPC algorithm schema shown in the section on the key ideas. The step "analyse" is crucial in the context of rewriting terms: before adjoining (s, t) to G it must be analysed whether $s > t$ or $t > s$ or s is incomparable with t. In the first two cases, (s, t) or (t, s) is adjoined respectively. In the third case the algorithm must be terminated "with failure". In general, nothing interesting can be said in this case. Termination of this algorithm can not be guaranteed in general. In the cases when it terminates, the resulting set G of identities has the property that $\rightarrow_G$ is Church–Rosser. The original correctness proof of the algorithm was already based on a version of Newman's lemma. Again, the crucial part of the proof, which concerns the power contained in the concept of critical pairs, goes beyond Newman's lemma and must also take account of the technicalities pertinent to first-order terms.

Completed sets: The notion of a "complete set" is explicitly introduced in (Knuth & Bendix, 1967) by essentially defining: G is complete iff $\rightarrow_G$ has the Church–Rosser property.

Remarks: The algorithm and the correctness proof presented in (Knuth & Bendix, 1967), actually, is due to Knuth alone, whereas Bendix, a student of Knuth, did the implementation as has been pointed out to me by D. Lankford. From the point of view of heuristics, it is interesting to see that, in the original paper (Knuth & Bendix, 1967), the authors do not mention the intuition of locating "the first possible superposition situation" as the view underlying the notion of a critical pair. Instead, they apparently arrived at the notion of a critical pair by a careful analysis of how local confluence can be injured, i.e. in which cases it is possible that, in a situation $s \leftarrow_G r \rightarrow_G t$, s and t have no common successor. It turns out that only one "critical case" remains, namely just the case of a general superposition of "spectra", which they found can be reduced to the case of a "most general superposition".

Introductory reading: Most probably Huet (1977) contains the best presentation of the Knuth–Bendix algorithm.

Early Forerunners

The three papers mentioned in the preceding section had a number of early forerunners that contained, more or less explicitly, the two key ideas of the CPC approach. In retrospect, by the recent increased interest in the CPC method, more and more of these forerunners are discovered. Of course, (Newman, 1942) is a basis for all considerations on establishing the Church–Rosser property. However, in this section, I would like to concentrate on forerunners that showed versions or initial ideas of the *critical-pair concept* which, personally, I consider as the crucial part of the CPC approach. (Newman's lemma does not turn the infinite completion procedure into a finite algorithm. Rather, it only may help to nicely organise correctness proofs. In fact, only one of the correctness proofs of the three papers discussed in the last section, namely (Knuth & Bendix, 1967) was based on Newman's lemma. Also, the idea of critical pairs may be useful in situations where the Church–Rosser property is not the main concern.)

FORERUNNERS OF THE RESOLUTION PROCEDURE

J. A. Robinson himself traces the idea of his resolution algorithm back to (Prawitz, 1960). In (Robinson, 1979, p. 292) he writes: "The idea that, instead of *trying* all

instantiations over the Herbrand Universe, one might *predict* which ones would produce a 'winning combination' by using what we have called the Unification Algorithm, turns out to have been sitting there all these years, unnoticed, in Herbrand's doctoral thesis (Herbrand, 1930). (Prawitz, 1960), following ideas of (Kanger, 1957), was, as far as the present author is aware, the first to describe this idea at length in print." At a different place, (Robinson, 1967), he writes: "The unification algorithm is essentially a cleaned-up and simplified version of the process described somewhat obscurely in (Prawitz, 1960). Recently it came to my attention that essentially the same procedure was found by the late Emil Post and called by him the 'LCM' process, but was never published (see Davis, 1965)." For me this latter reference is particularly thrilling because in my own algorithm I compute an LCM (without quotation marks) for obtaining a critical pair. Unification, of course, is a crucial ingredient in the first-order resolution algorithm. However, I think it is the combination of unification (looking for the "first possible", the "most general" interesting situation) *and* (propositional) resolution (reduction by cutting away the unified parts) that makes Robinson's algorithm a "critical pair" algorithm.

FORERUNNERS OF THE POLYNOMIAL IDEAL ALGORITHM

My own research on the polynomial ideal algorithm was stimulated by my thesis advisor, Prof. W. Gröbner (1899–1980). He encouraged me to work on the problem described in the preceding section (finding multiplication tables for residue class rings), which he presented in a seminar 1964 together with his own ideas on how to attack the problem. In the terminology of the section on key ideas, his approach was as follows: Consider all power products r and reduce them to normal form (i.e. compute s and t such that $\underline{s} \leftarrow_G^* r \rightarrow_G^* \underline{t}$) in all possible ways using the polynomials in G. If $s \neq t$ then adjoin $s-t$ to G. Thus, he proposed a completion procedure. However, he did not yet see that we can directly move to the "first possible" power products, where distinct reductions can occur, i.e. to the least common multiples of leading power products, and that it *suffices* to consider only these particular power products. Also, he did not really present a general correctness proof for his procedure. Still, he showed extremely sound intuition because he recommended starting systematically from low power products in terms of degree. Thus, he was very close to critical pairs.

In (Buchberger, 1970) I quoted Gröbner's ideas as an "oral communication". Strangely enough, Gröbner never told me that he had published his ideas already in 1950 (Gröbner, 1950) with the following additional remark: "I have used this method for approximately 17 years in various cases including complicated ones and . . .". Only by chance, in 1984, during a stay in Halle (GDR) I learned from B. Renschuch, who had written a book in the spirit of W. Gröbner's ideal theoretical approach to commutative algebra (Renschuch, 1976), about the existence of (Gröbner, 1950). When I resumed work on the subject of constructive methods for polynomial ideals in (Buchberger, 1976), in order to underline Gröbner's crucial contribution to my 1965 algorithm, I called the completed sets obtained by application of the algorithm "Gröbner bases" and gave various characterisations for them.

In fact, as pointed out in (Buchberger, 1965, 1970), in the case of univariate polynomials and in the case of linear multivariate polynomials my 1965 algorithm specialises to Euclid's algorithm and Gauss' algorithm respectively. Hence, these two algorithms may be viewed as very early forerunners.

Recently some authors, for example Bayer (1982), have also pointed out that essentially the same notion as "Gröbner bases" (i.e. the notion of bases with respect to which all

elements in an ideal can be reduced to zero), in the context of formal power series, had already been introduced in Hironaka's famous paper (Hironaka, 1964). Hironaka calls these ideal bases "standard bases" and calls the corresponding reduction process "division algorithm". He proves that for every ideal basis F there exists a corresponding standard basis. However, the proof is not constructive (it is essentially done by the "Schreier-construction", see (Bauer, 1981, p. 18)), i.e. it does not give an algorithm how to obtain the standard basis corresponding to a given arbitrary basis F. In particular, there is no indication of the idea of critical pairs in Hironaka's paper. A short version of such an inconstructive existence proof for Gröbner bases, from which Hilbert's basis theorem can be obtained as a corollary, is also presented in (Buchberger, 1982).

FORERUNNERS OF THE KNUTH–BENDIX ALGORITHM

In (Knuth & Bendix, 1965), both very general and very special credit is given to forerunners: "The formal development of this paper is primarily a precise statement of what hundreds of mathematicians have been doing for many decades, so no great claims of originality are intended for most of the concepts or methods used. .... The main new contribution of this paper is intended to be an extension of some methods used by Trevor Evans (Evans, 1951). We allow operators of arbitrary degree, and we make use of a well-ordering of words which allows us to treat axioms such as the associative law." Consulting (Evans, 1951) one sees that the core of the method to which Knuth alludes seems to be a procedure described in (Evans, 1951*a*, p. 69) by which a "closed" set of relations is produced from generators and relations of certain finitely generated algebras in the presence of equational axioms. When preparing this lecture. I tried to find out whether the procedure described in (Evans, 1951*a*), which surely has the character of a "completion" procedure, also contains the concept of "critical pairs". Frankly, I was not able to decide, and I think that, in any case, Knuth showed a lot of ingenuity when deriving the very general and clearly formulated procedure in (Knuth & Bendix, 1967) from (Evans, 1951 or 1951*a*). See, however, the preface of (Lankford, 1975) and (Lankford & Butler, 1984) for an appreciation of Evans' work as a forerunner of (Knuth & Bendix, 1967).

OTHER FORERUNNERS

In retrospect, it seems that several of the early algebraic algorithms had some flavor of the CPC method, see, for example (Dehn, 1911), (Greenlinger, 1960), and also Gauss' algorithm for presenting a symmetrical polynomial in terms of the elementary symmetrical polynomials, as has been pointed out by Loos (1981).

Generalisations, Independent Developments and Unifying Approaches

GENERALISATIONS OF THE RESOLUTION PRINCIPLE

It is not the objective of this paper to review how the resolution method developed since its invention in 1965. (For the various refinements of the method see the textbooks on automated theorem proving, for example, (Chang & Lee, 1973), (Loveland, 1978), (Robinson, 1979)). Here, we are exclusively interested in the CPC aspect of the resolution method. In this respect, paramodulation as introduced by (Robinson & Wos, 1969) can be viewed as a further inclusion of CPC ideas in resolution: for treating the equality sign,

instead of resolvents, paramodulants can be taken from two clauses. Roughly, paramodulation can be viewed as allowing resolution and forming critical pairs in the style of the Knuth–Bendix procedure at the same time. A step further in this direction was the introduction of "narrowing" in (Fay, 1979). One narrowing step transforming term t into term t' consists of three substeps:

a subterm of t is singled out that can be unified by a most general unifier σ with the left-hand side of a rule,
in $\sigma(t)$ the subterm is replaced by the right-hand side of the rule (after application of σ),
the resulting term is reduced to normal form using the rules of the rule system.

An overview on narrowing with applications is contained in (Rety *et al.*, 1985).

A more intimate amalgamation of the resolution method with the Knuth–Bendix completion procedure was introduced by (Lankford, 1975, 1975*a*), inspired by (Slagle, 1974) who considered term simplifiers without referring to the Knuth–Bendix procedure. Lankford proposed using the Knuth–Bendix procedure in connection with the resolution method in order to complete the sets of equalities used in the resolution method. In (Lankford & Ballantyne, 1979) it is shown that this leads to a refutation complete proof procedure that goes beyond paramodulation. The ideas developed in this approach also led to a recent interaction between resolution theorem proving and the Knuth–Bendix type completion procedure developed by J. Hsiang and others, see the section on the generalisations of the Knuth–Bendix procedure.

GENERALISATIONS OF THE CPC APPROACH FOR POLYNOMIAL RINGS

Lauer (1976) was the first to modify my 1965 algorithm to treat polynomial ideals with integer coefficients. The case of integer coefficients needs a modification of the reduction relation "$s \rightarrow_F t$" because, in one step, one cannot expect to be able to totally cancel a power product in s since, in the multipliers, we only have integer coefficients available. Instead of one type of S-polynomials ("critical pairs"), Lauer had to introduce two different types, "S-polynomials" and "T-polynomials". As the main result he proved that a finite set F of multivariate polynomials with integer coefficients is a Gröbner basis if the S-polynomials and T-polynomials of the elements in F can be reduced to zero modulo F. (I think that Lauer's work is important since being able to construct Gröbner bases for integer polynomial ideals implies that the uniform word problem for finitely generated rings can be solved.)

In this context it should be also mentioned that in 1976, based on the methods of Szekeres (1952), R. Stokhamer gave an algorithm for constructing "canonical forms" of multivariate integer polynomials modulo a given ideal F (see Stokhamer, 1975, 1976). A polished version of Stokhamer's work is also contained in (Lauer, 1976) and, in short version, in (Lauer, 1976*a*). Stokhamer's method is not a CPC method. It is still not sufficiently worked out how his method compares with the CPC methods, see however (Winkler, 1983, 1984).

A different generalisation of my algorithm was proposed in (Spear, 1977) and, independently, in (Trinks, 1978). Their approach consists in defining a class of rings that allow an algorithmic solution of the ideal membership and the syzygy problem (construction of generators for the solution set of linear diophantine equations) and in showing that if R belongs to this class then also $R[x_1, \ldots, x_n]$ belongs to that class. The proof is constructive and involves a generalisation of my 1965 algorithm. Instead of the

original critical pairs, combinations of the multiples of all the elements in F have to be taken whose leading monomials cancel. The approach of Spear (1977) was fully developed in (Zacharias, 1978) and also in (Schaller, 1979). In (Schaller, 1979) the rings satisfying the above effectiveness conditions are called "simplification rings". Since fields are simplification rings their approach yields my 1965 algorithm as a special case. Since Z is also a simplification ring, they also achieve a constructive method for obtaining Gröbner bases over Z (and, hence, solving the basic algorithmic problems for integer polynomial ideals).

Apparently independently of my own work, Bergman (1978) rediscovered essentially the same algorithm, however, in a slightly more general form, namely for free associative k-algebras $k\langle X\rangle$, where X is a set (of indeterminates) and k is a commutative, associative ring with 1. These algebras cover an impressively broad range. However, the approach is not broad enough to encompass the case of integer polynomial ideals because (Bergman, 1978) only admits pure words in X as the left-hand sides of "patterns" in F. A generalisation of my algorithm for the case of non-commutative polynomials has been announced in (Mora, 1985*a*).

In an independent effort, Ballantyne & Lankford (1981) also rediscovered a special case of my algorithm, namely the case when F contains only polynomials of the form $p-q$, where p and q are power products. These sets of polynomials may be viewed as the relations describing finitely generated abelian semigroups. The algorithm then yields a solution to the uniform word problem for finitely generated abelian semigroups. Although the algorithm in (Ballantyne & Lankford, 1981) is a special case of mine, it is interesting because it was the beginning of a merge of the two branches in CPC research stemming from my 1965 algorithm and from the Knuth–Bendix algorithm, see also the next subsection.

Yet another generalisation was initiated by Bayer (1982) and further developed by Mora & Möller (1983*a*) who consider $K[x_1, \ldots, x_n]$-modules and ideals in these modules instead of considering simply ideals in $K[x_1, \ldots, x_n]$. Early ideas in this direction were also announced by Guiver (1982). Mora (1985) also treats local rings by the same method. These generalisations produce important applications in algebraic geometry. A special algorithm patterned after my 1965 algorithm but with different term ordering was given for the computation of tangent cones (Mora, 1982).

Also working independently, Galligo became interested in standard bases for modules over $K[x_1, \ldots, x_n]$. After some earlier work, for example (Galligo, 1979), in which he did not consider the construction of standard bases but only the division algorithm with respect to standard bases, in (Galligo, 1984) he developed an algorithm for constructing standard bases that, again, could be viewed as a CPC algorithm. The same idea was then used in (Castro, 1984) for ideals of differential operators (see also Galligo, 1985).

Recently, I pursued a new axiomatic approach to generalising the CPC method to general rings, not only polynomial rings (see Buchberger, 1983*a*). As a byproduct, this approach yields a CPC algorithm for $Z[x_1, \ldots, x_n]$ whose structure is identical with my original algorithm for the case of field coefficients. It neither needs two different kinds of S-polynomials as in (Lauer, 1976) nor does the axiomatisation involve the relatively complicated conditions of the above simplification rings. Independently, arriving from studying the interplay between the Knuth–Bendix algorithm and my algorithm, in (Kandri-Rody & Kapur, 1984), for the special case of $Z[x_1, \ldots, x_n]$, essentially the same algorithm as in (Buchberger, 1983*a*) is developed and then generalised to the case of $R[x_1, \ldots, x_n]$ for Euclidean coefficient rings R in (Kandri-Rody & Kapur, 1984*a*).

GENERALISATIONS OF THE KNUTH–BENDIX ALGORITHM

The main direction in generalising the Knuth–Bendix algorithm was to establish procedures that can handle the case when some of the axioms in F destroy the finite termination property of the reduction relation. The general approach pursued for resolving this difficulty was to separate the set A of axioms into two groups of axioms, R and E, and to consider the axioms in R as generators of a corresponding reduction relation $\rightarrow_R$ whereas the axioms E are considered to generate a congruence relation $\sim_E$ on the set of terms. The problem then is to develop algorithms that, essentially, operate on the *congruence classes of terms* w.r.t. $\sim E$ rather than on the set of terms.

On the *set theoretical level* all these approaches are based on generalisations and refinements of Newman's lemma (Newman, 1942), which has been elegantly proven in (Huet, 1977) based on earlier work in (Church & Rosser, 1936; Hindley, 1969, 1974; Aho *et al.*, 1972; Sethi, 1974; Lankford, 1975; Staples, 1975; see Huet, 1977, for a detailed reference to these contributions). Recently, (Coquand & Huet, 1985) gave a machine-checked proof of Newman's lemma. In connection with working over equivalence classes, various generalised versions of Newman's lemma have been proven, for example in (Huet, 1977). (A generalisation of Newman's lemma of a totally different type, with a different purpose, is developed in (Buchberger, 1983*a*).)

In the context of generalising the Knuth–Bendix method for congruence classes of terms, *E-unification* (the generalisation of the original unification problem for E-congruence classes) plays an essential role. (Two terms s and t are E-unifiable iff there exists a substitution σ such that $\sigma s \sim_E \sigma t$.) E-unification was initiated in (Plotkin, 1972). A bibliography on E-unification is (Raulefs *et al.*, 1979). Extensive bibliographies are also contained in (Lankford, 1980) and (Fages, 1983). Most of the work on E-unification for different sets of axioms E, (including Huet, 1976; Livesey & Siekmann, 1976; Makanin, 1977; Siekmann, 1978; Lankford, 1979; Fay, 1979) is reviewed in (Huet & Oppen, 1980). Some recent papers on E-unification are (Siekmann & Szabo, 1982; Kirchner, 1984; Jouannaud *et al.*, 1983; Fages & Huet, 1983; Yellick, 1985; Fortenbacher, 1985; Tiden & Arnborg, 1985). Because of its practical importance, the case of E consisting of axioms expressing associativity and commutativity of function symbols was of central interest in E-unification research. Stickel (1975, 1976, 1981) developed an algorithm for generating a complete set of unifiers for the associative-commutative (AC) case and showed its partial correctness. However, only recently F. Fages (Fages, 1983, 1984) was able to show its total correctness by very subtle complexity measures for terms. This was one of the main achievements in unification research. As a subproblem of the AC-unification problem the solution of linear diophantine equations over the natural numbers appears. A crude algorithm for this problem in (Stickel, 1975) is improved in (Huet, 1978).

For attacking the problem of establishing a generalisation of the Knuth–Bendix method for quotient sets of terms modulo $\sim_E$, *different approaches* have been developed in the literature. For the discussion of these approaches let us define

$$\rightarrow_{R/E} := \sim_E \circ \rightarrow_R \circ \sim_E,$$

i.e. $\rightarrow_{R/E}$ is the reduction relation induced on the congruence classes modulo $\sim_E$.

The first approach is the one developed in (Lankford & Ballantyne, 1977*a*,*b*). It works with the above induced reduction relation. The original notion of critical pairs of terms is used for the representatives of the classes. This approach works for finite congruence classes only, because in general the induced reduction relation is undecidable for infinite

congruence classes. However, the method tends to be quite inefficient also in the limited case of finite congruence classes.

The other approaches may be viewed in one general framework by introducing one more reduction relation R' between $\rightarrow_R$ and $\rightarrow_{R/E}$:

$$\rightarrow_R \subset R' \subset \rightarrow_{R/E}.$$

This general view has been worked out in (Jouannaud & Kirchner, 1984). R' gives one more degree of freedom in executing reductions and establishing Church–Rosser theorems. Using R', the following notions may be distinguished (recall $A = E \cup R$):

$\rightarrow_R$ is R'-Church–Rosser (modulo E) iff
$t1 \sim_A t2$ implies
$t1 \rightarrow_{R'}{}^* t1'$, $t2 \rightarrow_{R'}{}^* t2'$ and $t1' \sim_E t2'$ (for some $t1'$, $t2'$),

$\rightarrow_{R'}$ is confluent (modulo E) iff
$t \rightarrow_{R'}{}^* t1$, $t \rightarrow_{R'}{}^* t2$ implies
$t1 \rightarrow_{R'}{}^* t1'$, $t2 \rightarrow_{R'}{}^* t2'$, $t1' \sim_E t2'$ (for some $t1'$, $t2'$),

$\rightarrow_R$ is R'-local-confluent (modulo E) iff
$t \rightarrow_{R'} t1$, $t \rightarrow_R t2$ implies
$t1 \rightarrow_{R'}{}^* t1'$, $t2 \rightarrow_{R'}{}^* t2'$, $t1' \sim_E t2'$ (for some $t1'$, $t2'$),

$\rightarrow_{R'}$ is coherent (modulo E) iff
$t \rightarrow_{R'}{}^+ t1$, $t \sim_E t2$ implies
$t1 \rightarrow_{R'}{}^* t1'$, $t2 \rightarrow_{R'}{}^+ t2'$, $t1' \sim_E t2'$ (for some $t1'$, $t2'$),

$\rightarrow_{R'}$ is local-coherent (modulo E) iff
$t \rightarrow_{R'} t1$, $t \sim_E t2$ implies
$t1 \rightarrow_{R'}{}^* t1'$, $t2 \rightarrow_{R'}{}^+ t2'$, $t1' \sim_E t2'$ (for some $t1'$, $t2'$).

Now the most general Church–Rosser result, due to (Jouannaud & Kirchner, 1984), can be formulated as follows:

$\rightarrow_R$ is R'-local-confluent and $\rightarrow_{R'}$ is local-coherent iff (*)
$\rightarrow_R$ is R'-Church–Rosser.

The link between $\rightarrow_{R'}$ and $\rightarrow_{R/E}$ is established by the following observation:

If $\rightarrow_{R/E}$ is terminating, $\rightarrow_{R'}$ is confluent and coherent then
$\rightarrow_{R'}$-normal-forms and $\rightarrow_{R/E}$-normal-forms coincide.

Hence, instead of working witch R/E, one may work with R' on representatives in the E-congruence classes. In retrospect, one now may view the different approaches as having worked with different R':

The approach initiated by Lankford & Ballantyne (1977*c*) and, independently, by Peterson & Stickel in a preliminary version of (Peterson & Stickel, 1981) uses $R' := R, E$ (i.e. matching modulo E) and a version of (*) that replaces coherence by compatibility (which is stronger). This approach is restricted to linear equations that possess a finite-complete E-unification algorithm. The approach developed in (Huet, 1977) uses $R' := R$. This approach is restricted to left-linear rules. (Pederson, 1984, 1985) use $R' := R \circ E$, where E-equalities are allowed only in variable substitutions, but not at internal occurrences of the rule to be applied. This permits to rewrite an instance of a left-hand

side with multiple occurrences of a variable having different (but E-equal) values. Based on (*), this approach has no theoretical restrictions. Jouannaud & Kirchner (1984) use $R' := Rl \cup Rnl, E$, where Rl contains only left-linear rules and Rnl contains the other rules. This approach is restricted to equational theories E with finite congruence classes and finite-complete unification algorithm. In Kirchner's thesis this approach is generalised to using $R' := Rl \cup Rnl1, E \cup Rnl2 \circ E$. In all these approaches, then, it must be shown that the left-hand side of (*) is implied by the confluence of critical pairs.

A CPC algorithm for the associative-commutative case is given in (Peterson & Stickel, 1981). This algorithm has been successfully applied to various axiom systems. A drawback of the algorithm is: it may add new rules even when the initial set of rules generates already a Church–Rosser reduction. Pederson (1984) gives an algorithm based on this approach. However, now proof details are provided. Huet (1977) gives an algorithm for left-linear rules. It is faster than the others, but may diverge in examples where the Peterson–Stickel algorithm would converge, because R as a rewrite relation is not strong enough in some practical cases. On the other hand, Huet's algorithm is a semi-decision procedure for equality, which is not guaranteed for the Peterson–Stickel algorithm. Jouannaud & Kirchner (1984) give an algorithm that combines advantages of Huet's and Peterson–Stickel's algorithm.

Other sources for improvements and generalisations of the above approaches (including the case when $\rightarrow_{R/E}$ is not terminating) are (Fages, 1983), (Padawitz, 1983), (Jouannaud *et al.*, 1983), (Perdrix, 1984), (Göbel, 1984).

Some research has also been carried out in generalising the Knuth–Bendix completion procedure to conditional rewrite systems (i.e. equations preceded by conditions as necessary, for example, for formulating certain of the field axioms). This research was initiated by (Lankford, 1979*a*,*c*), see also (Brand *et al.*, 1978). The first critical-pair result in this area was proved in (Remy, 1982). A recent paper is (Kaplan, 1984), which contains the references relevant for the Knuth–Bendix approach in conditional rewriting.

Recently, Lankford & Butler (1984), Ballantyne *et al.* (1984) and Butler & Lankford (1984) seem to develop a different approach to completing systems involving AC-axioms that resumes the early Evans approach of embedding and brings it together with the methods in (Buchberger, 1983*a*) and (Kandri-Rody & Kapur, 1984, 1984*a*) of constructing Gröbner bases for integer polynomial ideals. A complete analysis of the possible interactions between the approaches seems to be one of the most promising future research topics. It is exciting to see that the research activities that started twenty years ago from very different roots, finally, meet and merge: see also the next section.

THEOREM PROVING BY EQUATIONAL REWRITING AND POLYNOMIAL REDUCTION

The Peterson–Stickel methodology, by which complete sets of axioms can be derived for a wide class of equational theories, gave rise to an interesting connection between resolution theorem proving and the CPC procedures for term rewriting. Whereas in the early approach of Lankford (1975, 1975*a*) the completion procedure for equational axioms was embedded as a subalgorithm into the resolution procedure, in the recent approach by Hsiang (1981, 1982; see also Hsiang & Dershowitz, 1983) the resolution mechanism itself is described as a reduction with respect to an equational axioms system. The same approach has also been considered by Fages (1983) and, recently, by Paul (1985, 1985*a*). At the core of this method are complete axiom systems for boolean algebra. The existence of such systems is by no means trivial since the straightforward approach by

prime implicants does not lead to unique normal forms for boolean terms. Hsiang (1982) arrives at a complete system of axioms for boolean rings by using the "exclusive or" instead of the usual "or". Basically the canonical forms obtained are the Reed–Muller forms (Reed, 1954; Muller, 1954), although in these early papers no notion of term rewriting was involved. (Implicitly these forms are also contained in Stone's theorem on the representation of boolean algebras and in the Venn diagrams.) Roughly, the connection between the resolution method and equational rewriting is established by proving theorems of the following type

> A set C of clauses is unsatisfiable iff
> the Peterson–Stickel completion procedure applied to the boolean algebra axiom system together with an equational transcription of the clauses in C yields the equation "$1 = 0$".

Kapur & Narendran (1985) propose a similar approach that use the algorithm developed in (Buchberger, 1983*a*) and (Kandri-Rody & Kapur, 1984) for the completion of polynomial ideals over rings instead of the Peterson–Stickel algorithm.

In the above approach, equational rewriting is used for clausal theorem proving. However, it is conjectured to apply to non-clausal theorem proving as well. In this direction the recent work of Manna & Waldinger (1985) is of particular interest. They provide a class of inference rules for the treatment of special relations in automated deduction that are based on a general notion of polarity. The rules generalise the paramodulation and E-resolution rules to an arbitrary binary relation. I conjecture that the Manna & Waldinger (1985) approach and the above approach could eventually merge, if one extended the above approach to non-clausal theorem proving and, at the same time, views the rules in (Manna & Waldinger, 1985) as a very general critical-pair formation.

UNIFYING THE KNUTH–BENDIX ALGORITHM WITH THE GRÖBNER BASIS APPROACH

Motivated by the structural similarity between the Knuth–Bendix algorithm and my 1965 algorithm for constructing Gröbner bases, a number of people have tried to show that, in fact, my algorithm can be viewed as a special case of the Knuth–Bendix algorithm. Since my algorithm is concerned with operations in commutative fields (or rings) it is near at hand that embedding my algorithm in the Knuth–Bendix methodology can only be achieved by considering the Peterson–Stickel generalisation with AC-unification. Still, there is a crucial difficulty in the case of field coefficients because of the fact that the field axioms are not pure equations. Also, there is an essential difference between the variables used in first order term rewriting and the indeterminates in polynomial rings. The latter obstacle can be handled by considering the indeterminates as constants in the corresponding rewrite system. Starting from a rough sketch in (Loos, 1981) of how my algorithm could be embedded in the Peterson–Stickel procedure, the following papers were concerned with filling in the details: (Kandri-Rody & Kapur, 1983; Llopis de Trias, 1983; Le Chenadec, 1983). However, none of these papers could really close all the gaps. For an analysis of the subtle deficiencies left open in these papers see (Winkler, 1984), who also provided a construction of a completion procedure that incorporates the Peterson–Stickel procedure and my 1965 algorithm as special cases. The construction developed in (Winkler, 1984) distinguishes between reduction and simplification steps. This is an idea borrowed from (Kandri-Rody & Kapur, 1984).

Although such embeddings do not add to the computational efficiency of the algorithms (the more general an algorithm is the more it must leave out potential knowledge that can speed up the algorithm in the special case), they have some theoretical value for obtaining a clear picture of what is essential in the constructions. For practical purposes, however, it is of course much better to use the "least common multiple" construction of the polynomial ideal algorithms than to use the whole mechanism of AC-unification (and essentially arrive at the same end.) Embeddings, however, can have the practical significance of making flexible implementations possible that follow the software philosophy of polymorphic data types, see for example the new SCRATCHPAD system (Jenks, 1984).

Still, I think that a totally different approach based on an axiomatisation of the notion of patterns, multipliers and replacements should be tried sometime, not for obtaining good algorithms but out of structural interest. In this respect (Bauer, 1981) is an outstanding paper that developed much of the CPC approach from an axiomatisation of the concept of "substitution".

OTHER CPC APPROACHES

Loos (1981) gives some examples of algorithms that could be conceived as CPC algorithms, for example, the collection algorithm in computational group theory (McDonald, 1976). There is also a major research activity in confluent Thue systems, see Book (1982, 1982*a*, 1985). However, it seems that in this context the tool of critical pairs is not in the centre of interest (see, however, Nivat, 1971). Also, the unification algorithm used as a subalgorithm in resolution theorem proving and in the Knuth–Bendix procedure could be viewed as a CPC algorithm.

Applications

APPLICATIONS OF THE RESOLUTION METHOD

Since universal theorem proving theoretically provides the mechanisation of mathematics, the application of the resolution method would be universal. In practice, however, there are some reservations whether universal theorem proving will ever play an important role in mathematical discovery. Special theorem provers for particular decidable theories, as for example the quantifier elimination method in (Collins, 1975) seem to be more promising for eventual impact on mathematical research. Rather, I guess that most people will agree that universal theorem proving has had its biggest practical impact by providing a method for constructing terms that establish answers to existential theorems. Starting from early (1965) efforts using resolution theorem proving in "question answering" and "program deduction"—see Chang & Lee (1973) for a review—this development now becomes of central importance in the logic programming movement (Kowalski, 1979). A fair assessment of the ubiquitious applications of resolution theorem proving is beyond the expertise of the present author.

APPLICATIONS OF THE GRÖBNER BASES METHOD

A review of applications of the Gröbner basis method is given in Buchberger (1985). We summarise the most important achievements.

APPLICATIONS FOR PROPERTIES OF IDEALS

After having transformed an arbitrary polynomial ideal basis F to a corresponding Gröbner basis G, a number of important algorithmic problems can be easily solved: $f \in \text{Ideal}(F)$ can be decided by reducing f to normal form modulo G. Actually, the normal form algorithm with respect to G is a canonical simplifier for ideal congruence. As a special case, the uniform word problem for commutative semigroups (reachability problem for reversible Petri nets) can be solved. Furthermore, the structure of the residue class ring modulo Ideal(F) is fully available: a vector space basis and the complete multiplication table for the residue class ring can be found by the method sketched in the section on the mid sixties algorithms. An easy test on the leading power products of the polynomials in the Gröbner basis G shows whether the ideal has (topological) dimension zero or has higher dimension. Inverses of elements in the residue class rings can be calculated if they exist. (This has applications in radical simplification.) Furthermore, it is easy to compute the Hilbert function of an ideal when a Gröbner basis is known. The above applications are contained in the original paper (Buchberger, 1965). Schrader (1976) has shown how to apply Gröbner bases for supporting primary decomposition computations for polynomial ideals. Kandri-Rody (1984) has shown how, for Gröbner bases, the maximality and primality of polynomial ideals can be tested. He also gave a method for determining the radical and the dimension of an ideal. Gianni & Trager (1985) have given methods for factorisation and gcd computations for multivariate polynomials using Gröbner bases. Mora & Möller (1983) have given a better algorithm for computing the Hilbert-function bases on Gröbner bases. Mora (1982) gives a CPC algorithm for computing the tangent cone of a polynomial ideal. Möller (1976) and Möller & Buchberger (1982) apply Gröbner bases to construct certain desirable auxiliary formulae for designing multivariate numerical integration formulae. Wolf (1985) shows how systems of differential equations can be transformed algebraically to Gröbner basis form in order to facilitate subsequent numerical solution. Wu (1978), for geometrical theorem proving, considers polynomial sets that are close to the characteristic sets considered in (Kandri-Rody, 1984). Characteristic sets can be obtained easily from Gröbner bases. Geometrical theorem proving using Gröbner bases is considered in (Kutzler & Stifter, 1985). Robbiano & Valla (1983) consider the application of Gröbner bases to the set-theoretic complete intersection problem for curves in p^3.

The algorithm in (Buchberger, 1983*a*) and (Kandri-Rody & Kapur, 1984, 1984*a*), besides allowing solution of the membership and canonical simplification problem for polynomial ideals over the integers, also yields a solution for the uniform word problem for finitely presented commutative rings. (The defining relations can be viewed as polynomials with integer coefficients.) Since Gröbner bases are unique (Buchberger, 1976; Kandri-Rody & Kapur, 1984) one can construct bijective enumerations of all polynomial ideals over Z. This may lead to a classification of all possible residue class rings. A compilation of results on word problem for various finitely presented structures based on the Gröbner bases approach is given in (Kandri-Rody *et al.*, 1985). In this paper also some complexity results for the respective algorithms are given.

APPLICATION FOR THE EXACT SOLUTION OF POLYNOMIAL EQUATIONS

Having a Gröbner basis G corresponding to a set F of multivariate polynomials (equations) one can effectively determine univariate polynomials $p_i(x_i)$ with minimal degree in the ideal such that the set of zeros of $G(F)$ is contained in the combinations of

zeros of the p_i (Buchberger, 1970). This method has been improved in (Böge *et al.*, 1985) by combining it with factorisation. A different method based on Gröbner basis has been introduced in (Trinks, 1978). Trinks observed that if Gröbner bases are computed with respect to the "purely lexicographical term ordering" then they automatically turn out to have their variables "separated". Thus, the system of equations can be solved by successive computation of the components of the zeros. Stated differently, a Gröbner bases G w.r.t. the purely lexicographical ordering allows reading off of all the elimination ideals directly from the elements in G.

APPLICATION FOR COMPUTATION OF SYZYGIES

Spear (1977) and (Zacharias (1978) first observed that the S-polynomials (critical pairs) of the elements of a Gröbner bases essentially form a generating set for the module of all syzygies of a polynomial ideal, i.e. linear diophantine equations with polynomial coefficients can be solved effectively by considering S-polynomials. This method is easily extended to compute the whole chain of syzygy modules of a polynomial ideal (the "free resolution"), see Bayer (1982) and Mora & Möller (1983*a*). Möller (1985) has applied the method to Taylor resolution.

APPLICATIONS OF THE KNUTH–BENDIX TYPE ALGORITHMS

The original and main application of the Knuth–Bendix algorithm and the algorithms derived from it are the completion of equational axiom systems for solving the word problems for the respective equational theories, i.e. the problem to decide, for given terms s and t, whether "$s = t$" can be derived in the theory by equational reasoning. (It also should be stated here that by the correctness proof for the Knuth–Bendix procedure given in (Huet, 1981) the procedure, under certain conditions, yields at least a semidecision procedure for "$s = t$" even if it does not terminate). Through the years, by different variants of the procedure, an impressive list of axiom systems have been completed. Such lists have been published several times in the literature. Thus, in the present paper, we only point to these reviews: (Knuth & Bendix, 1967; Lankford, 1979*b*, 1980; Hullot, 1980; Butler & Lankford, 1980; Le Chenadec, 1983; Jouannaud & Kirchner, 1984). As pointed out earlier, recently a particular interest evolved in completing boolean algebra (Hsiang, 1982; Fages, 1983; Paul, 1985). For some more examples see (Pederson, 1984).

For some important axiom systems—lattices and modular lattices, Heyting algebras, Lie algebras, fields—no complete systems are known.

Special completion methods have been derived for solving word problems for finitely generated groups, see (Bücken, 1979, 1979*a*) and (Richter & Kemmenich, 1980), and also (Le Chenadec, 1983, 1985). The word problem for a finitely presented group may be solved by deciding whether or not a given word is equal to the unit element. This idea leads to the notion of limited confluence (which is useful also in boolean algebras, see the work of Paul (1985) on the confluence of valid formulae). Dehn's algorithm (Dehn, 1911) for small cancellation groups is a well known algorithm that follows this idea. This algorithm may be viewed as a rewriting system computed by completion limited to superpositions between a finite group presentation and the complete sets of rules for free groups. The connection between Dehn's algorithm and completion algorithms was proved in Bücken's thesis and further analysed by Le Chenadec. Bauer (1981) analyses connections between rewrite rules and standard combinatorial techniques (free, direct, semi-direct and amalgamated products etc.) and gives a complexity hierarchy. Le

Chenadec gives complete systems for several families of classical groups (Coxeter, polyhedral, surface groups etc.) See also (Lankford *et al.*, 1983) for the case of finitely presented abelian groups.

Choppy & Johnen (1985) give applications of completed rule sets for deciding properties of Petri nets. (In the case of reversible Petri nets the reachability property is known to be easily decidable using the method of Gröbner bases, see (Buchberger, 1985)).

Particular practical interest in completed axiom systems developed from progress in the use of abstract data types. For practical computation in the direct implementation of algebraically specified abstract data types the decidability of the corresponding word problem for terms is a prerequisite, see for example (Gerhart *et al.*, 1980; Lichtenberger, 1980). The Knuth–Bendix completion method provides a general tool for producing decision algorithms. Research on this application was pursued, for example, by Musser (1977, 1978, 1978*a*), Goguen *et al.* (1982), Musser & Kapur (1982) and Jouannaud (1983).

An exciting application was developed by Musser (1980), Huet & Hullot (1980), Goguen (1980) and others: "inductionless induction". The method consists of showing the validity of an equation e in the "initial model" of a set E of equations by applying the Knuth–Bendix completion algorithm to $E \cup \{e\}$. (Under certain conditions) e is valid in the initial model iff the algorithm stops without generating an inconsistency ($1 = 0$). Thus, this method allows us to automatically prove sentences that normally would require invention of an induction hypothesis for an inductive proof. The original method is due to Musser with extensions by Goguen. They require that the user axiomatises equality. The method by Huet & Hullot does not require this but presupposes that the set of function symbols is partitioned into constructors (completely free operators) and other operators. An extension of the method that works under very weak assumptions is announced by Jouannaud & Kounalis. Finally, the application of equational completion for simulating clausal theorem proving may be viewed as a step in the same direction but more general in nature. The respective contributions were reviewed in the section on generalisations.

Technicalities

TERMINATION

Termination of Gröbner bases algorithms. Reduction of polynomials is always noetherian. Thus, the termination problem in Gröbner basis algorithms is mainly the problem of terminating completion. For my 1965 algorithm the termination proof for the general case is given in (Buchberger, 1970), where Dickson's lemma has been rediscovered. Essentially Dickson's lemma is also sufficient for showing termination of other Gröbner basis algorithms. A very elegant proof of Dickson's lemma, based on abstract properties of products of ordered set, has recently been given by Cousineau (1984).

LEMMA (*Dickson*, 1983)*: A sequence of n-tuples* $e_1, e_2, \ldots$ *of non-negative integers, such that in* e_i *at least one component is greater than in* e_j *when* $i < j$*, is finite.*

A different termination proof for my algorithm can be given by applying Hilbert's basis theorem, see (Bergman, 1978). Alternatively, Dickson's lemma and the concept of a

Gröbner basis can be taken as primitives and Hilbert's basis theorem, then, is a corollary, see (Buchberger, 1982). The equivalence of Dickson's lemma and Hilbert's basis theorem, together with other interesting results, were also proven in (Butler & Lankford, 1984). Kollreider (1978) and Robbiano (1985) characterised the term orderings that are admissible for the Gröbner basis approach.

Termination of Knuth–Bendix type algorithms. In this context, the crucial problem is the invention of noetherian term orderings that are compatible with the reduction. Several methods have been developed: well-founded mapping, increasing interpretation, simplification ordering, recursive path ordering, homomorphic interpretation etc. The literature on this topic, including (Kruskal, 1960; Nash-Williams, 1963; Manna & Ness, 1970; Dershowitz, 1979*a,b*; Dershowitz & Manna, 1979; Plaisted, 1978, 1978*a*; Lankford, 1979*b*; Kamin & Lévy, 1980 and others) is reviewed in (Huet & Oppen, 1980). Recent contributions to termination orderings are (Jeanrond, 1980; Lescanne, 1981; Dershowitz, 1982; Jouannaud *et al.*, 1982; Jouannaud & Lescanne, 1982; Munoz, 1983; Plaisted, 1983; Lescanne, 1984; Jouannaud & Munoz, 1984; Bachmair & Plaisted, 1985).

Huet & Lankford (1978) show that the uniform halting problem for rewrite systems is undecidable.

Strategies

Strategies for the resolution method. The sequence in which resolvents are built and other techniques play a crucial role for improving the efficiency of the resolution method. This was an extensive field for research; see the textbooks on resolution theorem proving mentioned above. The work of Hsiang sketched above may also be seen in this context. His *N*-strategy can be viewed as a combination of parts of several strategies used in resolution theorem proving. In the context of this paper the strategies used in resolution theorem proving are interesting because, by the similarities between the CPC algorithms, it might be worthwhile to study possibilities for carrying some of the strategies from the resolution method over to, say, the Knuth–Bendix type algorithms. This idea has been pursued to a certain extent in (Küchlin, 1982).

Strategies for Knuth–Bendix and Gröbner bases algorithms. Already in (Buchberger, 1965) and in (Knuth–Bendix, 1967) it was suggested to keep the systems of patterns developing in the course of the algorithm mutually reduced. The organisation of this strategy is a non-trivial task. Huet (1981) gave a complete proof for the correctness of the Knuth–Bendix algorithm when incorporating successive reduction of rules. In the case of Gröbner bases algorithms, in addition, it is suggested to consider first critical pairs with overlapping power products of low degree. The reasons for this are explained in some detail in (Buchberger, 1979).

Criteria for Omitting Certain Critical Pairs

In (Buchberger, 1979) it was shown that certain critical pairs need not be considered in the CPC algorithm because it can be predicted that they are reducible to the same normal form. The theoretical reason for this can be based on a generalised Newman lemma that has been formulated in full generality in (Buchberger, 1983*a*). I included its proof in (Winkler & Buchberger, 1983). Roughly, the lemma says that for guaranteeing local confluence it suffices that each critical pair can be connected "below" its common ancestor. Recently, it has been shown that essentially the same criterion can be applied to Knuth–Bendix type algorithms (Winkler & Buchberger, 1983; Winkler, 1985; Küchlin,

1985). The approach of speeding up CPC algorithms by criteria of the above type is based on additional mathematical insight rather than on heuristics.

COMPLEXITY

Relatively little is known on the complexity of CPC algorithms.

Complexity of Gröbner bases algorithms. A first (very coarse) complexity analysis for the case of two variables was already given in (Buchberger, 1965). A realistic analysis for this case can be found in (Buchberger, 1983) and for the trivariate case in (Winkler, 1984, 1984*a*). In the bivariate case the maximal degree of the Gröbner basis corresponding to a given basis is a linear function of the maximal degree of the input polynomials. A similar bound is derived in (Lazard, 1983) and in (Giusti, 1985). Lazard showed that a linear bound also holds in the general case of arbitrarily many variables if some special properties of the ideal are presupposed. Similar results are contained in (Bayer, 1982). Möller & Mora (1984) give some more bounds on the degrees. Intrinsically, the problem of computing Gröbner bases is difficult. Cardoza *et al.* (1976) and Mayr & Meyer (1981) showed that the uniform membership problem for polynomial ideals is essentially exponentially space complete. This is supported by the investigation in (Huynh, 1984) for some special classes. The practical relevance of these results is a matter of point of view. (Applications of CPC algorithms often are in the "sweep coherence" (Goad, 1980) context). However, practical experiences show that, in fact, Gröbner bases computations are complex. Polynomials in six variables of degree, say, three mark the size of problems that are at the border of practical feasibility.

Complexity of Knuth–Bendix computations. The only complexity result I am aware of is (Bauer & Otto, 1984), where it is shown that complete rewrite systems may have an arbitrarily complex word problem.

IMPLEMENTATIONS

Implementations of the resolution method. A review on the many implementations of the resolution method is beyond the expertise of the present author. See the textbooks on automated theorem proving.

Implementations of Gröbner basis algorithms. Special implementations have been undertaken several times: (Buchberger, 1965; Schrader, 1976; Zacharias, 1978; Winkler, 1978; Schaller, 1979). A very flexible and user-friendly implementation is described in (Böge *et al.*, 1985). This implementation is in the SAC-2 computer algebra system (Collins & Loos, 1980). At present, implementations are also available in all other major computer algebra systems, for example, MACSYMA (Pavelle & Wang, 1985), REDUCE (Hearn, 1984), NEW SCRATCHPAD (Jenks, 1984, muMATH (Stoutemyer, 1985).

Implementations of Knuth–Bendix type algorithms. The first implementation was in (Knuth & Bendix, 1967). Other implementations in "rewrite rule laboratories" are described in (Hullot, 1980; Stickel, 1981; Küchlin, 1982*a*; Lescanne, 1983; Kirchner & Kirchner, 1983; Kapur & Sivakumar, 1983; Le Chenadec, 1983; Forgaard & Guttag, 1984; Thomas, 1984; Dick, 1985; Hussmann, 1985; Musser, 1980*a*) in the framework of AFFIRM and (Futatsugi *et al.*, 1985) in the framework of OBJ. Fages (1985) is a manual on the (Hullot, 1980) completion system developed at INRIA. A symposium on implementations of the Knuth–Bendix algorithm has been held recently (Guttag *et al.*, 1984). Stickel's implementation is now available on LISP machines and seems to be the fastest implementation for full CA. Hullot's system, the KB system, is well documented

and easily available (in MacLISP, FranzLISP, ZetaLISP etc.). The REVE system (Lescanne, 1983; Forgaard & Guttag, 1984; Kirchner & Kirchner, 1983, 1985) provides semi-automated termination proofs.

Note added September 1986*:* Meanwhile a number of additional papers on the critical-pair/completion approach have been published. Many of them are contained in the recent issues of the *Journal of Symbolic Computation* or will be included in forthcoming issues. Some of them are contained in the Proceedings of the ACM 1986 Symposium on Symbolic and Algebraic Computation (Bruce W. Char, ed.), available from ACM. Recently Le Chenadec's thesis appeared in book form (Le Chenadec, 1986). It emphasises the study of particular finitely presented algebras but contains also a survey of the critical-pair completion approach and an elaborate list of references.

J. Dick, F. Fages, J. Hsiang, G. Huet, J.-P. Jouannaud, S. Kaplan, P. Le Chenadec, and D. Musser gave valuable comments on various parts of the paper. J. Dick did a careful proof reading. My best thanks to all of them. This work was supported by a grant from SIEMENS Munich (Dr H. Schwärtzel).

References

Aho, A., Sethi, R., Ullman, J. D. (1972). Code Optimisation and finite Church–Rosser theorems. In: (Rustin, R. ed.) *Design and Optimisation of Compilers.* pp. 89–105. Englewood Cliffs: Prentice Hall.

Armbruster, D. (1985). Bifurcation theory and computer algebra: an initial approach. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 126–137.

Bachmair, L., Buchberger, B. (1980). A simplified proof of the characterisation theorem for Gröbner bases. *ACM SIGSAM Bull.* **14,** (4), 29–34.

Bachmair, L., Plaisted, D. A. (1985). Termination orderings for associative commutative rewriting systems. *J. Symbolic Computation* **1,** 329–350.

Ballantyne, A. M., Lankford, D. S. (1981). New decision algorithms for finitely presented commutative semigroups. *Computers and Maths. with Appls.* **7,** 159–165: preliminary version: Rep. MTP-4, Louisiana Tech Univ., Dep. of Math. 1979.

Ballantyne, M., Butler, G., Lankford, D. (1984). *On practical uniform decision algorithms for the uniform word problem in finitely presented commutative rings of characteristics* $\infty, \ldots, 3, 2, 1$. Tech. Report, Louisiana Tech. Univ., Dep. of Math.

Bauer, G. (1981). *The representation of monoids by confluent rule systems.* Ph.D. thesis, University of Kaiserlautern (FRG), Dept of Comp. Sci.

Bauer, G., Otto, F. (1984). *Finite complete rewriting systems and the complexity of the word problem.* Manuscript, Univ. of Kaiserslautern, Dep. Comp. Scie.

Bayer, D. (1982). *The division algorithm and the Hilbert scheme.* Ph.D. thesis, Harvard University, Cambridge, Mass., Math. Dept.

Bergman, G. M. (1978). The diamond lemma for ring theory. *Adv. Matn.* **29,** 178–218.

Blass, A. Gurevich, Yu. (1983). Equivalence relations, invariants, and normal forms. Proc. Logic and Machines: Decision Problems and Complexity, (ed. by E. Börger, G. Hasenjaeger, D. Rödding), *Springer Lec. Notes Comp. Sci.* **171,** 24–42.

Böge, W., Gebauer, R., Kredl, H. (1985). Gröbner-Bases using SAC2. *Proc. EUROCAL 85.*

Book, R. V. (1982). Confluent and other types of Thue systems. *J. Assoc. Comp. Mach.* **29,** 171–182.

Book, R. V. (1982*a*). The power of the Church–Rosser property for string rewriting systems. Proc. 6th Conference on Automated Deduction, New York, (ed. by D. W. Loveland), *Springer Lec. Notes Comp. Sci.* **138,** 360–368.

Book, R. V. (1985). Thue systems as rewriting systems. (This issue.)

Brandt, D., Darringer, J. A., Joyner, W. H. (1978). *Completeness of conditional reductions.* Yorktown Heights: IBM Research Center.

Buchberger, B. (1965). *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (German).* Ph.D. thesis, Univ. of Innsbruck (Austria), Math. Inst.

Buchberger, B. (1970). An algorithmical criterion for the solvability of algebraic systems of equations (German). *Aequ. Math.* **4,** 374–383.

Buchberger, B. (1976). A theoretical basis for the reduction of polynomials to canonical form. *ACM SIGSAM Bull.* **10/3,** 19–29; **10/4,** 19–24.

Buchberger, B. (1979). A criterion for detecting unnecessary reductions in the construction of Gröbner bases. Proc. EUROSAM 79, Marseille, June 1979, (ed. by W. Ng), *Springer Lec. Notes Comp. Sci.* **72,** 3–21.

Buchberger, B. (1982). Miscellaneous Results on Gröbner bases for polynomial ideals II. Techn. Report CAMP 82–23, Univ. of Linz, Math. Inst.; also Techn. Report 83–1, Univ. of Delaware, Dep. of Comp. and Inform. Scie.

Buchberger, B. (1983). A note on the complexity of constructing Gröbner bases. Proc. EUROCAL 83, London, March 1983, (ed. by J. A. van Hulzen), *Springer Lec. Notes Comp. Sci.* **162,** 137–145.

Buchberger, B. (1983*a*). A critical-pair/completion algorithm for finitely generated ideals in rings. Proc. Logic and Machines: Decision Problems and Complexity, (ed. by E. Börger, G. Hasenjaeger, D. Rödding), *Springer Lec. Notes Comp. Sci.* **171,** 137–161.

Buchberger, B. (1985). Gröbner bases: an algorithmic method in polynomial ideal theory. Chapter 6. In: (Bose, N. K., ed.) *Multidimensional Systems Theory*. Dordrecht: Reidel.

Buchberger, B., Loos, R. (1982). Algebraic simplification. In: (Buchberger, B., Collins, G., Loos, R., ed.) *Computer Algebra—Symbolic and Algebraic Computation* pp. 11–43. Vienna: Springer.

Bücken, H. (1979). *Reduction systems and word problems (German)*. Rep. 3, RWTH, Aachen, FRG, Computer Science Inst.

Bücken, H. (1979). Reduction systems and small cancellation theory. *Proc. 4th Workshop on Automated Deduction,* 53–59.

Butler, G., Lankford, D. (1980). *Experiments with computer implementations of procedures which often derive decision algorithms for the word problem in abstract algebras.* Techn. Rep. MTP-7, Louisiana Tech. Univ., Dep. of Math.

Butler, G., Lankford, D. (1984). *Dickson's lemma, Hilbert's basis theorem, and applications to completion in commutative noetherian rings.* Manuscript, Dept. Math. and Statistics, Louisiana Tech. Univ., Ruston.

Cardoza, E., Lipton, R., Meyer, A. R. (1976). Exponential space complete problems for Petri nets and commutative semigroups. *Conf. Record of the 8th Annual ACM Symp. on Theory of Computing,* 50–54.

Castro, F. (1984). *Théorème de division pour les opérateurs différentiels et calcul des multiplicités.* Thèse 3ème cycle, Univ. Paris VII.

Chang, C., Lee, R. C. (1973). *Symbolic logic and mechanical theorem proving*. New York: Academic Press.

Church, A., Rosser, J. B. (1936). Some properties of conversion. *Trans. Amer. Math. Soc.* **39,** 472–482.

Collins, G. (1975). Quantifier elimination for real closed fields by cylindrical algebraic decomposition. Proc. 2nd GI Conf. on Automata Theory and Formal Languages, (ed. by H. Brakhage), Kaiserslautern, *Springer Lec. Notes Comp. Sci.* **33,** 134–183.

Collins, G. E., Loos, R. G. (1980). ALDES and SAC-2 now available. *ACM SIGSAM Bull.* **14/2,** 19.

Coquand, T., Huet, G. (1985). Constructions: a higher order proof system for mechanising mathematics. Invited paper in Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **203,** 151–185.

Cousineau, G. (1984). *Preuves de terminaison des systèmes de réécriture.* Notes de cours de DEA, University Paris 7.

Davis, M. (ed.) (1965). *The undecidable: basic papers on undecidable propositions, unsolvable problems and computable functions.* New York: Raven Press.

Dehn, M. (1911). On infinite discontinuous groups (German). *Math. Ann.* **71,** 116–144.

Dershowitz, N. (1979*a*). Orderings for term-rewriting systems. *Proc. 20th Symp. on Foundations of Comp. Sci.* 123–131.

Dershowitz, N. (1979*b*). A note on simplification orderings. *Inf. Proc. Lett.* **9/5,** 212–215.

Dershowitz, N. (1982). Orderings for term-rewriting systems. *J. Theor. Comp. Sci.* **17,** 279–301.

Dershowitz, N. (1982*a*). *Applications of the Knuth–Bendix completion procedure.* Prelim. rep., Univ. of Illinois at Urbana-Champaign, Dep. Comp. Sci.

Dershowitz, N. (1986). Termination issues in term rewriting systems. (This issue.)

Dershowitz, N., Manna, Z. (1979). Proving termination with multiset orderings. *Commun. ACM* **22,** 465–475.

Dershowitz, N., Hsiang, J., Josephson, N. A., Plaisted, D. A. (1983). Associative-commutative rewriting. *Proc. 10th IJCAI,* Karlsruhe, 1983, 990–994.

Detlefs, D., Forgaard, R. (1985). A procedure for automatically proving the termination of a set of rewrite rules. *Springer Lec. Notes Comp. Sci.* **202,** 255–270.

Dick, A. J. J. (1985). ERIL—Equational reasoning: an interactive laboratory. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 400–401.

Dickson, L. E. (1913). Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *Am. J. Math.* **35,** 413–426.

Evans, T. (1951). On multiplicative systems defined by generators and relations I: normal form theorems. *Proc. Cambridge Philosophy Soc.* **47,** 637–649.

Evans, T. (1951*a*). The word problem for abstract algebras. *J. Lond. Math. Soc.* **26,** 64–71.

Evans, T. (1978). Word problems. *Bull. Amer. Math. Soc.* **84/5:** 789–802.

Fages, F. (1983). *Formes canoniques dans les algèbres booléennes et applications à la démonstration automatique en logique de premier ordre.* Thèse 3ème cycle, Universite Paris VI.

Fages, F. (1984). Associative-commutative unification. Proc. 7th CADE, (ed. by R. Shostak), Napa Valley. *Springer Lec. Notes Comp. Sci.* **170.**

Fages, F. (1985). *KB reference manual.* INRIA Rep. no. 368.
Fages, F., Huet, G. (1983). Complete sets of unifiers and matchers in equational theories. Proc. CAAP 83, L'Aqila, (ed. by G. Ausiello, M. Protasi),. *Springer Lec. Notes Comp. Sci.* **159,** 205–220.
Fay, M. (1979). First order unification in equational theories. Proc. 4th CADE. *Springer Lec. Notes Comp. Sci.* **87,** 161–167.
Forgaard, R., Guttag, J. V. (1984). REVE: A term rewriting system generator with failure-resistant Knuth–Bendix. MIT-LCS.
Fortenbacher, A. (1986). An algebraic approach to unification under associativity and commutativity. *J. Symbolic Computation* **3,** (to appear).
Futasugi, K., Goguen, J., Jouannaud, J. P., Meseguer, J. (1985). Principles of OBJ2. *Proc. of the 1985 Symp. on Principles of Progr. Lang.*
Galligo, A. (1979). The division theorem and stability in local analytic geometry (French). *Extrait des Annales de l'Institut Fourier, Univ. of Grenoble* **29/2.**
Galligo, A. (1984). *Algorithmes de calcul de base standards.* Manuscript, Univ. of Nice, France.
Galligo, A. (1985). Some algorithmic questions on ideals of differential operators. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 413–421.
Gerhart, S. L., Musser, D. R., Thompson, D. H., Baker, D. A., Bates, R. L., Erickson, R. W., London, R. L., Taylor, D. G., Wile, D. S. (1980). An overview on AFFIRM: a specification and verification system. In: (Lavington, S. H., ed.) *Information Processing 80* pp. 343–387. Amsterdam: North-Holland.
Gianni, P., Trager, B. (1985). GCD's and factoring multivariate polynomials using Gröbner bases. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 409–410.
Giusti, M. (1984). Some effectivity problems in polynomial ideal theory. Proc. EUROSAM 84, Cambridge, (ed. J. Fitch). *Springer Lec. Notes Comp. Sci.* **174,** 159–172.
Giusti, M. (1985). A note on the complexity of constructing standard bases. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 411–412.
Goad, C. A. (1980). Proofs of descriptions of computation. Proc. 5th CADE. *Springer Lec. Notes Comp. Sci.* **87,** 39–52.
Göbel, R. (1983). *A completion procedure for globally finite term rewriting systems.* Proc. of an NSF Workshop on the Rewrite Rule Laboratory, General Electric, 1983, Schenectady, (ed. by J. V. Guttag, D. Kapur, D. R. Musser), Rep. no. 84GEN008, 155–206.
Goguen, J. A. (1980). How to prove algebraic inductive hypotheses without induction, with applications to the correctness of data type implementations. Proc. 5th CADE, (ed. by W. Bibel and R. Kowalski), *Springer Lec. Notes Comp. Sci.* **87,** 356–373.
Goguen, J. A., Meseguer, J., Plaisted, D. (1982). Programming with parameterised abstract objects in OBJ. In: (Ferrari, D., Bolognani, M., Goguen, J., eds) *Theory and Practice of Software Technology,* pp. 163–193. Amsterdam: North Holland.
Greenlinger, M. (1960). Dehn's algorithm for the word problem. *Comm. Pure Appl. Math.* **13,** 67–83.
Gröbner, W. (1950). On elimination theory (German). *Monatshefte für Mathematik* **54,** 71–78.
Guiver, J. P. (1982). *Contributions to two-dimensional systems theory.* Ph.D. thesis, Univ. of Pittsburgh, Math. Dept.
Guttag, J. V., Kapur, D., Musser, D. R. (eds) (1984). *Proc. of an NSF Workshop on the Rewrite Rule Laboratory,* Sept. 6–9, 1983. General Electric, Schenectady, NY, Rep. no. 84GEN008.
Hearn, A. C. (1984). Reduce user's manual: version 3.1. Santa Monica, California: The Rand Corporation.
Herbrand, J. (1930). Researches in the theory of demonstration. Ph.D. thesis, Univ. of Paris, Reprinted in: (Heijenoort, J. van, ed.) *From Frege to Gödel: A Source Book in Mathematical Logic,* Cambridge, Mass: Harvard University Press.
Hermann, G. (1926). The question of finitely many steps in the theory of polynomial ideals (German). *Math. Ann.* **95,** 736–788.
Hindley, R. (1969). An abstract form of the Church–Rosser theorem I. *J. Symb. Logic* **34/4,** 545–560.
Hindley, R. (1974). An abstract form of the Church–Rosser theorem II: applications. *J. Symb. Logic* **39/1,** 1–21.
Hironaka, H. (1964). Resolution of singularities of an algebraic variety over a field of characteristic zero: I, II. *Ann. Math.* **79,** 109–326.
Hsiang, J. (1981). *Refutational theorem proving using term rewriting systems.* AI J. 1985. Manuscript, Univ. of Illinois at Urbana Champaign, Dep. Comp. Sci.
Hsiang, J. (1982). *Topics in automated theorem proving and program generation.* AI J. 1985. Ph.D. thesis, Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Sci.
Hsiang, J. (1985). Rewrite methods for theorem proving in first order theory with equality. (This issue.)
Hsiang, J., Dershowitz, N. (1983). Rewrite methods for clausal and non-clausal theorem proving. Proc. ICALP 83, 10th Colloquium, Barcelona, Spain. *Springer Lec. Notes Comp. Sci.* **154,** 331–346.
Hsiang, J., Plaisted, D. (1982). *Deductive program generation.* Techn. Rep. Univ. of Illinois at Urbana-Champaign, Comp. Scie. Dept.
Huet, G. (1976). *Résolution d'équation dans des langages d'ordre 1, 2, . . ., ω.* Thèse d'état, Univ. Paris VII.
Huet, G. (1980). Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comp. Mach.* **27,** 797–821.

Huet, G. (1978). An algorithm to generate the basis of solutions to homogeneous linear diophantine equations. *Inf. Proc. Lett.* **7/3,** 144–147.

Huet, G. (1981). A complete proof of the Knuth–Bendix completion algorithm. *J. Comp. Syst. Sci.* **23,** 11–21.

Huet, G., Hullot, J. M. (1980). Proofs by induction in equational theories with constructors. *21st IEEE Symp. on Foundations of Comp. Sci.*, 96–107. Also: *J. Assoc. Comp. Mach.* **25,** (1982), 239–266.

Huet, G., Lankford, D. S. (1978). On the uniform halting problem for term rewriting systems. *Rapport Laboria* **283.** INRIA, Rocquencourt, Le Chesnay, France.

Huet, G., Oppen, D. C. (1980). Equations and rewrite rules: a survey. In: (Book, R. V., ed.) *Formal Language Theory, Perspectives and Open Problems.* 349–405. London: Academic Press.

Hullott, J. M. (1979). Associative-commutative pattern matching. *5th IJCAI*, Tokyo.

Hullot, J. M. (1980). *A catalogue of canonical term rewriting systems.* Techn. Rep. CSL-113, SRI International, Menlo Park, Calif.

Hullot, J. M. (1980*a*). Canonical forms and unification. Proc. 5th CADE, (ed. by W. Bibel and R. Kowalski). *Springer Lec. Notes Comp. Sci.* **87,** 318–334.

Hullot, J. M. (1980*b*). *Compilation de formes canoniques dans les théories équationnelles.* Thèse 3ème cycle, U. Paris Sud.

Hussmann, H. (1985). Unification in conditional equational theories. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 543–553.

Huynn, D. T. (1984). *The complexity of the membership problem for two subclasses of polynomial ideals.* Manuscript, Iowa, State Univ., Ames, Comp. Scie. Dep.

Jeanrond, J. (1980). Deciding unique termination of permutative rewrite systems: choose your term algebra carefully. Proc. 5th CADE, (ed. by W. Bibel and R. Kowalski). *Springer Lec. Notes Comp. Sci.* **87.**

Jenks, R. D. (1984). A primer: 11 keys to NEW SCRATCHPAD. Proc. EUROSAM 84, Cambridge, England, (ed. by J. Fitch). *Springer Lec. Notes Comp. Sci.* **174,** 123–147.

Jouannaud, J. P. (1983). Confluent and coherent equational term rewriting systems. Proc. CAAP 83, L'Aqila, (ed. by G. Ausiello, M. Protasi). *Springer Lec. Notes Comp. Sci.* **159,** 269–283.

Jouannaud, J. P., Kirchner, H. (1984). Completion of a set of rules modulo a set of equations. Proceedings 11th POPL, *SIAM J. Comput.* (to apear).

Jouannaud, J. P., Kirchner, C., Kirchner, H. (1983). Incremental construction of unification algorithms in equational theories. Proc. ICALP 83, 10th Colloquium, Barcelona, Spain. *Springer Lec. Notes Comp. Sci.* **154,** 361–373.

Jouannaud, J. P., Kirchner, H., Remy, J. L. (1983). Church–Rosser properties of weakly terminating equational term rewriting systems. *Proc. 10th IJCAI*, Karlsruhe.

Jouannaud, J. P., Lescanne, P., Reinig, F. (1982). Recursive decomposition ordering. *Conf. on Formal Description of Programming Concepts,* (D. Bjorner). Amsterdam: North Holland.

Jouannaud, J. P., Lescanne, P. (1982). On multiset orderings. *Inf. Proc. Lett.* **15,** 57–62.

Jouannaud, J. P., Munoz, M. (1984). Termination of a set of rules modulo a set of equations. Proc. 7th CADE, (ed. by R. Shostak). *Springer Lec. Notes Comp. Sci.* **170,** 175–193.

Kamin, Lévy (1980). *Attempts for generalising the recursive path ordering.* Unpublished manuscript, Univ. of Paris 7.

Kandri-Rody, A. (1984). *Effective methods in the theory of polynomial ideals.* Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY, Math. Dep.

Kandri-Rody, A., Kapur, D. (1983). *On relationship between Buchberger's Gröbner basis algorithm and the Knuth–Bendix completion procedure.* Rep. no. 83CRD286, General Electric, Schenectady.

Kandri-Rody, A., Kapur, D. (1984). Computing the Gröbner basis of an ideal in polynomial rings over the integers. *3rd MACSYMA Users' Conf.*, Schenectady, NY, July 1984.

Kandri-Rody, A., Kapur, D. (1984*a*). Algorithms for computing Gröbner bases of polynomial ideals over various Euclidean rings. Proc. EUROSAM 84, Cambridge, (ed. J. Fitch). *Springer Lec. Notes Comp. Sci.* **174,** 195–206.

Kandri-Rody, A., Kapur, D., Narendran, P. (1985). An ideal-theoretic approach to word problems and unification problems over finitely presented commutative algebras. Proc. RTA 85. *Springer Lec. Notes Comp. Sci.* **202,** 345–364.

Kanger, S. (1957). *Provability in logic.* Stockholm.

Kaplan, S. (1984). *Fair conditional rewriting systems: unification, termination and confluence.* Techn. Rep. No. 194, Lab. de Recherche en Informatique, Orsay, France.

Kapur, D., Krishnamurthy, B. (1983). A natural proof system based on rewriting techniques. *Proc. of an NSF Workshop on the Rewrite Rule Laboratory.* General Electric, Schenectady, 1983, (ed. by J. V. Guttag, D. Kapur, D. R. Musser), Rep. no. 84GEN008, 337–348.

Kapur, D., Narendran, P. (1985). An equational approach to theorem proving in first-order predicate calculus. *Proc. IJCAI 1985.* Los Angeles.

Kapur, D., Sivakumar, G. (1983). Architecture of and experiments with RRL, a rewrite rule laboratory. *Proc. of an NSF Workshop in the Rewrite Rule Laboratory.* General Electric, Schenectady, (ed. by J. V. Guttag, D. Kapur, D. R. Musser), Rep. no. 84GEN008, 33–56.

Kirchner, C. (1984). A new equational unification method: a generalisation of Martelli–Montanari algorithm. Proc. 7th CADE, (ed. by R. Shostak), *Springer Lec. Notes Comp. Sci.* **170.**

Kirchner, C., Kirchner, H. (1983). *Current implementation of the general E-completion algorithm.* Techn. Rep., Centre de Recherches en Informatique de Nancy.

Kirchner, H. (1984). A general inductive completion algorithm and application to abstract data types. Proc. 7th CADE, (ed. by R. Shostak). *Springer Lec. Notes Comp. Sci.* **170.**

Kirchner, C., Kirchner, H. (1985). Implementation of a general completion procedure parameterised by built-in theories and strategies. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 402–404.

Knuth, D. E. (1970). Notes on central groupoids. *J. Comb. Theory* **8,** 376–390.

Knuth, D. E., Bendix, P. B. (1967). Simple word problems in universal algebras. *Proc. of the Conf. on Computational Problems in Abstract Algebra,* Oxford, 1967, (ed. by J. Leech), Pergamon Press, Oxford, 1970, 263–298.

Kollreider, C. (1978). *Polynomial reduction: the influence of the ordering of terms on a reduction algorithm.* Techn. Rep. CAMP 78-4, Univ. of Linz, Austria (Europe), Math. Dep.

Kruskal, J. B. (1960). Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture. *Trans. Amer. Math. Soc.* **95,** 210–225.

Küchlin, W. (1982). A theorem-proving approach to the Knuth–Bendix completion algorithm. Proc. EUROCAM 82, Marseille, (ed. by J. Calmet). *Springer Lec. Notes Comp. Sci.* **144,** 101–108.

Küchlin, W. (1982*a*). *An implementation and investigation of the Knuth–Bendix completion procedure.* Internal Rep., Comp. Scie. Dept., Univ. of Karlsruhe, FRG.

Küchlin, W. (1985). A confluence criterion based on the generalised Newman lemma. Proc. EUROCAL 85, *Springer Lec. Notes Comp. Sci.* **204,** 390–399.

Kutzler, B., Stifter, S. (1985). *Geometrical theorem proving: a comparison between Wu's algorithm and the use of Gröbner bases.* Tech. Rep. CAMP 85-17.0, Univ. of Linz, Math. Dept.

Lankford, D. S. (1975). *Canonical algebraic simplification.* Rep. ATP-25, Univ. of Texas, Austin, Dep. Math. Comp. Sci.

Lankford, D. S. (1975*a*). *Canonical inference.* Rep. ATP-32, Univ. of Texas, Austin, Dep. Math. Comp. Sci.

Lankford, D. S. (1979). *A unification algorithm for abelian group theory.* Rep. MTP-1, Louisiana Tech Univ., Math. Dep.

Lankford, D. S. (1979*a*). *Mechanical theorem proving in field theory.* Rep. MTP-2, Louisiana Tech Univ., Math. Dep.

Lankford, D. S. (1979*b*). *On proving term rewriting systems are noetherian.* Rep. MTP-3, Louisiana Tech Univ., Math. Dep.

Lankford, D. S. (1979*c*). Some new approaches to the theory and applications of conditional term rewriting systems. Rep. MTP-6, Louisiana Tech Univ., Math. Dep.

Lankford, D. S. (1980). *Research in applied equational logic.* Rep. MTP-15, Louisiana Tech Univ., Ruston, Math. Dep.

Lankford, D. S. (1981). *A simple explanation of inductionless induction.* Rep. MTP-14, Louisiana Tech Univ., Ruston, Math. Dep.

Lankford, D. S., Ballantyne, A. M. (1977*a*). *Decision procedures for simple equational theories with commutative axioms: Complete sets of commutative reductions.* Rep. ATP-35, Univ. of Texas, Austin: Dep. Math. Comp. Sci.

Lankford, D. S., Ballantyne, A. M. (1977*b*). *Decision procedures for simple equational theories with permutative axioms: Complete sets of permutative reductions.* Rep. ATP-37, Univ. of Texas, Austin: Dep. Math. Comp. Sci.

Lankford, D. S., Ballantyne, A. M. (1977*c*). *Decision procedures for simple equational theories with commutative-associative axioms: Complete sets of commutative-associative reductions.* Rep. ATP-39, Univ. of Texas, Austin: Dep. Math. Comp. Sci.

Lankford, D. S., Ballantyne, A. M. (1979). The refutation completeness of blocked permutative narrowing and resolution. *4th CADE,* Austin, 53–59.

Lankford, D. S., Butler, G. (1984). *On the foundations of applied equational logic.* Tech. Rep., Louisiana Tech Univ., Dep. of Math.

Lankford, D. S., Butler, G. (1984). *On faster Smith normal form algorithms.* Manuscript, Louisiana Tech Univ., Dep. of Math.

Lankford, D. S., Butler, G., Ballantyne, A. M. (1983). A progress report on new decision algorithms for finitely presented abelian groups. *Proc. of an NSF Workshop on the Rewrite Rule Laboratory,* General Electrics Schenectady, 1983, (ed. by J. V. Guttag, O. Kapur, D. R. Musser), Rep. no. 84GEN008, 137–154.

Lauer, M. (1976). *Canonical representatives for residue classes of a polynomial ideal.* Diploma thesis, Univ. of Kaiserslautern, Dep. Math.

Lauer, M. (1976*a*). Canonical representatives for residue classes of a polynomial ideal. *Proc. ACM SYMSAC 76.* Yorktown Heights, NY, (ed. by R. D. Jenks), 339–345.

Lazard, D. (1982). Commutative algebra and computer algebra. Proc. EUROCAM 82. Marseille, France, (ed. by J. Calmet). *Springer Lec. Notes Comp. Sci.* **144,** 40–48.

Lazard, D. (1983). Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. Proc. EUROCAL 83. London, (ed. by J. A. van Hulzen). *Springer Lec. Notes Comp. Sci.* **162,** 146–156.

Le Chenadec, P. (1983). *Formes canoniques dans les algèbres finiment présentées.* Thèse 3ème cycle, Univ. Paris-Sud, Centre d'Orsay.

Le Cenadec, P. (1985). A Knuth–Bendix completion of some Coxeter groups. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 229–242.

Le Chenadec, P. (1986). *Canonical Forms in Finitely Presented Algebras.* (In the Research Notes in Theoretical Computer Science Series.) London: Pitman, and New York: J. Wiley.

Lescanne, P. (1981). Decomposition orderings as a tool to prove the termination of rewriting systems. *Proc. 7th IJCAI.* Vancouver, Canada, 548–550.

Lescanne, P. (1983). Computer experiments with the REVE term rewriting system generator. Proc. 10th POPL conference, Austin, Texas.

Lescanne, P. (1984). How to prove termination? An approach to the implementation of a new recursive decomposition ordering. *Proc. 6th Conf. on Automata, Algebra and Programming.* Bordeaux.

Lichtenberger, F. (1980). *PL/ADT: a system for using algebraically specified abstract data types* (German). Ph.D. thesis, Univ. of Linz, Austria (Europe), Math. Inst.

Lipton, R., Snyder, L. (1977). On the halting problem of tree replacement systems. *Conf. on Theoretical Computer Science.* U. of Waterloo, 43–46.

Livesey, M., Siekmann, J. (1976). *Unification of bags and sets.* Techn. Report, Univ. of Karlsruhe, FRG, Comp. Sci. Dep.

Llopis de Trias, R. (1983). *Canonical forms for residue classes of polynomial ideals and term rewriting systems.* Techn. Rep., Univ. Autónoma de Madrid, División de Matemática.

Loos, R. (1981). Term reduction systems and algebraic algorithms. Proc. GWAI-81, Bad Honnef, (ed. by J. H. Siekmann). *Springer Informatik-Fachberichte* **47,** 214–234.

Loveland, D. W. (1978). *Automated theorem proving: a logical basis.* Amsterdam: North-Holland.

MacDonald, I. D. (1974). A computer application to finite *p*-groups *J. Austral. Math. Soc.* **17,** 102–112.

Makanin, G. S. (1977). The problem of solvability of equations in a free semigroup. *Dokl. AN SSSR* **233/2.**

Manna, Z., Ness, S. (1970). On the termination of Markov algorithms. *3rd Hawaii Internat. Conf. on System Scie* 789–792.

Manna, Z., Waldinger, R. (1985). Special relations in automated deduction. Proc. of the 12th ICALP, Nafplion (Greece), *Springer Lec. Notes Comp. Sci.* **226.**

Mayr, E. W., Meyer, A. R. (1981). *The complexity of the word problems for commutative semigroups and polynomial ideals.* Report LCS/TM-199, M.I.T., Laboratory of Computer Science.

Metivier, B. (1983). About the rewriting systems produced by the Knuth–Bendix completion algorithm. *Inf. Proc. Lett.* **16,** 1983.

Möller, H. M. (1976). Multidimensional Hermite-interpolation and numerical integration. *Math. Z.* **148,** 107–118.

Möller, H. M. (1985). A reduction stretegy for the Taylor resolution. Proc. EUROCAL 85, *Springer Lec. Notes Comp. Sci.* **204,** 526–534.

Möller, H. M., Buchberger, B. (1982). The construction of multivariate polynomials with preassigned zeros. Proc. EUROCAM 82, Marseille, (ed. by J. Calmet). *Springer Lec. Notes Comp. Sci.* **144,** 24–31.

Möller, H. M., Mora, F. (1984). Upper and lower bounds for the degree of Gröbner bases. Proc. EUROSAM 84, Cambridge, (ed. by J. Fitch). *Springer Lec. Notes Comp. Sci.* **174,** 172–183.

Mora, F. (1982). An algorithm to compute the equations of tangent cones. Proc. EUROCAM 82. Marseille, (ed. J. Calmet). *Springer Lec. Notes Comp. Sci.* **144,** 158–165.

Mora, F. (1985). An algorithmic approach to local rings. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 510–525.

Mora, F. (1985*a*). *Gröbner bases for non-commutative polynomial rings.* AAECC-3 conference, Grenoble, July 1985, *Springer Lec. Notes Comp. Sci.* **229,** 353–362.

Mora, F., Möller, H. M. (1983). The computation of the Hilbert function. Proc. EUROCAL 83, London, (ed. by J. A. van Hulzen), *Springer Lec. Notes Comp. Sci.* **162,** 157–167.

Mora, F., Möller, H. M. (1983*a*). *New constructive methods in classical ideal theory.* Manuscript, Univ. of Genova, Italy, Math. Dept.; to appear in J. of Algebra.

Muller, D. E. (1954). Application of Boolean algebra to switching circuit design and error detection. *Trans. Inst. of Radio Eng.*, EC-3, 6–12.

Munoz, M. (1983). *Problème de terminaison finie des systèmes de réécriture équationels.* Thèse 3ème cycle, Univ. Nancy 1.

Musser, D. R. (1977). *A data type verification system based on rewrite rules.* Tech. Rep., USC Information Scie. Inst., Marina del Rey, Calif.

Musser, D. R. (1978). *Convergent sets of rewrite rules for abstract data types.* Tech. Rep., USC Information Scie. Inst., Marina del Rey, Calif.

Musser, D. R. (1978*a*). *A data type verification system based on rewrite rules.* 6th Texas Conf. on Computing Systems.

Musser, D. R. (1980). On proving inductive properties of abstract data types. *7th ACM Symp. POPL,* 154–162.

Musser, D. R. (1980*a*). Abstract data type specification in the AFFIRM system. *IEEE Trans. on S.E.,* SE-6, No. 1.

Musser, D. R., Kapur, D. (1982). Rewrite rule theory and abstract data type analysis. Proc. EUROCAM 82, Marseille, France, (ed. by J. Calmet). *Springer Lec. Notes Comp. Sci.* **144,** 77–90.

Nash-Williams, C. St. J. A. (1963). On well-quasi-ordering finite trees. *Proc. Cambridge Phil Soc.* **59**, 833–835.

Newman, M. H. A. (1942). On theories with a combinatorial definition of "equivalence". *Ann. Math.* **43/2**, 223–243.

Padawitz, P. (1983). Equational data type specification and recursive program schema. *IFIP Working Conf. on Formal Description of Programming Concepts II*, (ed. by D. Bjorner). Amsterdam: North-Holland.

Paul, E. (1985). Equational methods in first order predicate calculus. *J. Symbolic Computation* **1**, 7–30.

Paul, E. (1985*a*). On solving the equality problem in theories defined by Horn clauses. Proc. EUROCAL 85, *Springer Lec. Notes Comp. Sci.* **204.**

Pavelle, R., Wang, P. S. (1985). MACSYMA from F to G. *J. Symbolic Computation* **1**, 69–100.

Pederson, J. (1984). *Confluence methods and the word problem in universal algebra.* Ph.D. thesis, Emory Univ., Dep. Math. and Comp. Scie.

Pederson, J. (1985). Obtaining complete sets of reductions and equations without using special unification algorithms. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204**, 422–423.

Perdrix, H. (1984). Propriétés Church–Rosser de systèmes de réécriture équationnels ayant la propriété de terminaison faible. Proc. STACS 84, Paris, (ed. M. Fontet, K. Mehlhorn). *Springer Lec. Notes Comp. Sci.* **166**, 97–108.

Peterson, G. E. (1983). A technique for establishing completeness results in theorem proving with equality. *SIAM J. Comput.* **12**, 82–100.

Peterson, G. E., Stickel, M. E. (1981). Complete sets of reductions for some equational theories. *J. Assoc. Comp. Mach.* **28**, 233–264.

Plaisted, D. A. (1978). *Well-founded orderings for proving termination of systems of rewrite rules.* Rep. 78-932, Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Science.

Plaisted, D. A. (1978*a*). *A recursively defined ordering for proving termination of term rewriting systems.* Rep. 78-943, Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Science.

Plaisted, D. A. (1983). An associative path ordering. *Proc. of an NSF Workshop on the Rewrite Rule Laboratory.* General Electrics, Schenectady, (ed. by J. V. Guttag, D. Kapur, D. R. Musser), Rep. no. 84GEN008, 123–136.

Plotkin, G. (1972). Building-in equational theories. *Machine Intell.* **7**, 73–90.

Prawitz, D. (1960). An improved proof procedure. *Theoria* **26**, 102–139.

Raulefs, P., Siekmann, J., Szabó, P., Unvericht, E. (1979). A short survey on the state of the art in matching and unification problems. *ACM SIGSAM Bull.* **13/2**, 14–20.

Reed, I. S. (1954). A class of multiple error correcting codes and the decoding scheme. *Trans. Inst. Radio Eng.*, IT-4, 38–49.

Remy, J.-L. (1982). *Etude des systèmes de réécriture conditionnels et applications aux types abstraits algébriques.* Thèse d'état, Nancy, France.

Renschuch, B. (1976). *Elementary and practical ideal theory* (German). VEB Deutscher Verlag der Wissenschaften, Berlin.

Rety, P., Kirchner, C., Kirchner, H., Lescanne, P. (1985). Narrower, a new algorithm for unification and its application to logic programming. Proc. RTA 85, *Springer Lec. Notes Comp. Sci.* **202**, 141–157.

Richter, M. M., Kemmenich, S. (1980). *Reduction systems and decision procedures* (German). Rep. 4, RWTH, Aachen, FRG, Comp. Sci. Inst.

Robbiano, L. (1985). Term orderings on the polynomial ring. Proc. of the EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204**, 513–517.

Robbiano, L., Valla, G. (1983). *Some curves in P^3 are set-theoretic complete intersections.* Manuscript, Univ. of Genova, Italy, Dep. Math.

Robinson, G. A. Wos, L. T. (1969). Paramodulation and theorem proving in first-order theories with equality. *Machine Intell.* **4**, 135–150.

Robinson, J. A. (1963). A machine-oriented logic (abstract). *J. Symb. Logic* **28**, 302.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *J. Assoc. Comp. Mach.* **12/1**, 23–41.

Robinson, J. A. (1967). A review of automatic theorem proving. *Proc. Symp. Appl. Math., Am. Math. Soc.* **19**, 1–18.

Robinson, J. A. (1979). *Logic form and function. The mechanisation of deductive reasoning.* Edinburgh: University Press.

Rosen, B. K. (1971). *Subtree replacement systems.* Ph.D. thesis, Harvard Univ.

Rosen, B. K. (1973). Tree-manipulation systems and Church–Rosser theorems. *J. Assoc. Comp. Mach.* **20**, 160–187.

Rusinovitch, M. (1985). Path of subterms ordering and recursive decomposition ordering revisited. This issue.

Schaller, S. (1979). *Algorithmic aspects of polynomial residue class rings.* Ph.D. thesis, Techn. Rep 370, Univ. of Wisconsin-Madison, Comp. Scie. Dept.

Schrader, R. (1976). *Contributions to constructive ideal theory* (German). Diploma thesis, Univ. of Karlsruhe, FRG, Math. Inst.

Sethi, R. (1974). Testing for the Church–Rosser property. *J. Assoc. Comp. Mach.* **21/4**, 671–679.

Shtokhamer, R. (1975). *Simple ideal theory: some applications to algebraic simplification.* Tech. Rep. UCP-36, Univ. of Utah. Salt Lake City.

Shtokhamer, R. (1976). *A canonical form of polynomials in the presence of side relations.* Tech. Rep., Technion, Haifa, Phys. Dep.

Siekmann, J. (1978). *Unification and matching problems.* Ph.D. thesis, Memo CSM-4-78, University of Essex.

Siekmann, J., Szabo, P. (1981). *A Noetherian and confluent rewrite system for idempotent semigroups.* SEKI-project memo, Univ. of Karlsruhe, FRG, Dep. Comp. Scie.

Siekmann, J., Szabo, P. (1982). Universal unification and classification of equational theories. Proc. 6th CADE (ed. by D. W. Loveland). *Springer Lec. Notes Comp. Sci.* **138**.

Slagle, J. R. (1974). Automated theorem proving for theories with simplifiers, commutativity and associativity. *J. Assoc. Comp. Mach.* **21/4,** 622–642.

Smith, D. (1966). A basis algorithm for finitely generated Abelian groups. *Math. Algorithms* **1/1,** 13–26.

Spear, D. (1977). A constructive approach to commutative ring theory. *Proc. MACSYMA Users' Conf.* Berkeley, July 1977, (ed. by R. J. Fateman), published by M.I.T., 369–376.

Staples, J. (1975). Church–Rosser theorems for replacement systems. In: Algebra and Logic, (ed. by J. Crossley). *Springer Lec. Notes Math.* **450,** 291–307.

Stickel, M. E. (1975). A complete unification algorithm for associative-commutative functions. Advance papers *4th Int. Joint Conf. on Artificial Intelligence.* Tbilisi, USSR, pp. 71–76.

Stickel, M. E. (1977). *Unification algorithms for artificial intelligence languages.* Ph.D. thesis, Carnegie-Mellon Univ.

Stickel, M. E. (1981). A unification algorithm for associative commutative functions. *J. Assoc. Comp. Mach.* **28/3,** 423–434; preliminary version: 4th IJCAI, 1975.

Stoutemyer, D. R. (1985). A preview of the next IBM-PC version of muMATH. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **203,** 33–44.

Szekeres, G. (1952). A canonical basis for the ideals of a polynomial domain. *Am. Math. Monthly* **59/6,** 379–386.

Thomas, C. (1984). *RRLab—Rewrite Rule Labor.* Memo SEKI-84-01, Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, D6750 Kaiserslautern.

Tiden, E., Arnborg, S. (1985). Unification problems with one-sided distributivity. This issue.

Trinks, W. (1978). On B. Buchberger's method for solving systems of algebraic equations (German). *J. Number Theor.* **10/4,** 475–488.

Winkler, F. (1978). *Implementation of an algorithm for constructing Gröbner bases* (German). Diploma thesis, Univ. of Linz, Austria (Europe), Math. Inst., to appear in ACM TOMS.

Winkler, F. (1983). An algorithm for constructing detaching bases in the ring of polynomials over a field. Proc. EUROCAL 83, London, (ed. by J. A. van Hulzen). *Springer Lec. Notes Comp. Sci.* **162,** 168–179.

Winkler, F. (1984). *The Church–Rosser property in computer algebra and special theorem proving: an investigation of critical-pair/completion algorithms.* Ph.D. thesis, Univ. of Linz, Austria (Europe), Math. Inst.

Winkler, F. (1984*a*). On the complexity of the Gröbner-bases algorithm over $K[x, y, z]$. Proc. EUROSAM 84, Cambridge, (ed. by J. Fitch). *Springer Lec. Notes Comp. Sci.* **174,** 184–194.

Winkler, F. (1985). Reducing the complexity of the Knuth–Bendix completion algorithm: a "unification" of different approaches. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 378–389.

Winkler, F., Buchberger, B. (1983). A criterion for eliminating unnecessary reductions in the Knuth–Bendix algorithm. *Proc. of the Coll. on Algebra, Combinatorics and Logic in Comp. Scie. Györ, Coll. Math. Soc. J. Bolyai* **42,** 849–869.

Wolf, T. (1985). Analytic decoupling, decision of compatibility and partial inegration of systems of nonlinear ordinary and partial differential equations. Proc. EUROCAL 85. *Springer Lec. Notes Comp. Sci.* **204,** 517–598.

Wu, W. (1978). On the decision problem and the mechanisation of theorem-proving in elementary geometry. *Scientia Sinica* **21/2,** 159–172.

Yelick, K. (1985). Combining unification algorithms for confined regular equational theories. This issue.

Zacharias, G. (1978). *Generalised Gröbner bases in commutative polynomial rings.* Bachelor Thesis, M.I.T., Dept. Comp. Scie.