MIT/LCS/TR-268

DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ANALYZING TERM REWRITING SYSTEMS

John V. Guttag

Deepak Kapur

David R. Musser

DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ANALYZING TERM REWRITING SYSTEMS

John V. Guttag[1]

MIT Laboratory for Computer Science

Deepak Kapur

General Electric Research and Development Center

David R. Musser

General Electric Research and Development Center

Abstract

Starting from the seminal work of Knuth and Bendix, we develop several notions useful in the study of term rewriting systems. In particular we introduce the notions of "derived pairs" and "overlap closure" and show that they are useful in analyzing sets of rewrite rules for various properties related to termination. We also introduce a new representation, based on rewrite dominoes, for rewrite rules and sequences of rewrites.

Keywords: Confluence, Critical pair, Derived pair, Knuth-Bendix, Overlap closure, Restricted termination, Rewrite dominoes, Term rewriting systems, Uniform termination

1. Work partially supported by the National Science Foundation under grant MCS78-01798 and by an Office of Naval Research Contract with DARPA funding #N00014-75-C-0661.

1. INTRODUCTION

We introduce three new tools for the study of term rewriting systems. Derived pairs of a rewrite rule generalize the well-known idea of "critical pairs" introduced by Knuth and Bendix (1970) in their development of a method of proving the confluence property. The overlap closure of a set of rules is a set of rules that corresponds to a subset of the transitive closure of the rewriting relation. Its construction is based on the use of derived pairs obtained from superpositions of the right hand side of one rule with the left hand side of another. This process is closely related to the Knuth-Bendix process, which uses critical pairs for generating new rules in an attempt to achieve confluence. We use the overlap closure in proving - or disproving - that a rewriting relation is uniformly terminating.[1] It thus provides an interesting dual method to the Knuth-Bendix process, in which the validity of the critical pair test for confluence depends upon uniform termination. The combination of uniform termination and confluence provides a decision procedure for the theory of the equations corresponding to the original rules.

In the study of derived pairs and overlap closures we found it useful to devise a new way of representing rewrite rules and sequences of rewrites using what we call rewrite dominoes and "rewrite domino layouts". We will introduce this representation and use it in presenting the proofs of our main results about the overlap closure. We believe that this representation also will be useful in the study of other areas of rewrite rule theory.

Like the Knuth-Bendix process, the overlap closure process may fail to terminate (that is, it may continue to generate new rules indefinitely). In fact, when the original rules are uniformly terminating, it will usually happen that overlap closure generation is nonterminating. In this case, the overlap closure process does not by itself yield a proof of uniform termination, but it may be useful as an aid in applying other known methods of proving uniform termination [see Huet and Oppen, 1980]. It can also be used in proving what we call "restricted termination," i.e., termination for all terms up to a given size. Some applications of restricted termination are discussed in [Guttag, Kapur and Musser, 1981].

Perhaps more important is the case where the original rules are not uniformly terminating. One would often like to be able to detect this situation quickly, e.g., in order to avoid wasting time attempting to construct a proof of uniform termination. We show that under some reasonable restrictions on the form of rewrite rules, the overlap closure construction provides such a test.

1. more commonly called finitely terminating or noetherian.

I.e., we show that if the rules are globally finite (that is to say, the number of different terms to which any term can be rewritten is finite) and every rule is right-linear or every rule is left-linear, the overlap closure construction can be used to effectively search for cycles in the rewriting relation. (That it does so "quickly" enough to be useful is a claim for which we have limited empirical evidence, as discussed in the Conclusion section).

2. DEFINITION OF OVERLAP CLOSURE

For the most part we use standard definitions and terminology for term rewriting systems from Huet (1980) and Huet and Oppen (1980). There are a few exceptions, such as "uniform termination" for "finite termination," and "terminal form" for "normal form." In [Guttag, Kapur, and Musser, 1981], the reader will find a thorough discussion of this background material. Here we confine ourselves mainly to the definitions of "derived pairs," a generalization of the Knuth and Bendix's notion of "critical pairs," and of "overlap closure."

Two terms are said to overlap if one is unifiable with a nonvariable subterm of the other. If s and t overlap, we define their superposition: either

a) s unifies with a nonvariable subterm t' of t, by the most general unifier (m.g.u.) θ, in which case $\theta(t)$ is called a superposition of s and t; or

b) a nonvariable subterm s' of s unifies with t, by m.g.u. θ, in which case $\theta(s)$ is called a superposition of s and t.

Now consider ordered pairs of terms (r,s) and (t,u) such that s and t overlap, as above. (If the variables of t must be renamed, the same renaming must be applied to u.) Then along with the superposition $\theta(t)$ or $\theta(s)$ we obtain the derived pair of terms, $\langle p,q \rangle$, where

a) if s unifies with a nonvariable subterm t/i by m.g.u. θ,

p=[θ(t) with θ(r) at i][1]

q=θ(u);

b) if a nonvariable subterm s/i unifies with t by m.g.u. θ,

p=θ(r)

q=[θ(s) with θ(u) at i].

In the case of a rewriting system $\underline{R} = \{(l_i \rightarrow r_i)\}$, the derived pairs obtained from the pairs (r_i,l_i) and (l_j,r_j)) are called critical pairs.

1. The notation [t with u at i] stands for the term obtained from t by replacing the subterm at position i by u. A "subterm position" and "corresponding subterm" within a term is a finite sequence of nonnegative integers separated by "." and a related term determined as follows: to the null sequence (denoted $\langle\rangle$) corresponds the entire term. If $f(t_1,\ldots,t_n)$ is the subterm at position i then the subterm at position i.j is t_j. We write t/i for the subterm at position i within term t.

Consider, for example, obtaining a critical pair from the rewrite rules:

$$x^{-1} \bullet x \rightarrow e$$

$$(x' \bullet y') \bullet z' \rightarrow x' \bullet (y' \bullet z')$$

We begin by constructing the ordered pairs $(e, x^{-1} \bullet x)$ and $((x' \bullet y') \bullet z', x' \bullet (y' \bullet z'))$. Now $x^{-1} \bullet x$ can be unified with $x' \bullet y'$ using the substitution $\theta = [x^{-1}/x', x/y']$. This leads to the derived pair $\langle e \bullet z', x^{-1} \bullet (x \bullet z')\rangle$ which is a critical pair of the rules.

Using derived pairs, the <u>overlap closure of</u> $\underline{R}$, written $OC(\underline{R})$, is defined inductively as follows:

a. Every rule $r \rightarrow s$ in $\underline{R}$ is also in $OC(\underline{R})$.
b. Whenever $r \rightarrow s$ and $t \rightarrow u$ are in $OC(\underline{R})$, every derived pair $\langle p,q\rangle$ of (r,s) and (t,u) is in $OC(\underline{R})$ (as $p \rightarrow q$).
c. No other rules are in $OC(\underline{R})$.

"Examples of overlap closures:"

i. Let $\underline{R} = \{f(x) \rightarrow g(x)\}$, then $OC(\underline{R}) = \underline{R}$.
ii. Let $\underline{R} = \{f(x) \rightarrow g(h(x)), h(x) \rightarrow k(x)\}$, then $OC(\underline{R}) = \underline{R} \cup \{f(x) \rightarrow g(k(x))\}$.
iii. Let $\underline{R} = \{x \bullet (y \bullet z) \rightarrow (x \bullet y) \bullet z\}$, then from the superposition $(x \bullet (x' \bullet y')) \bullet z'$ we obtain the rule

$$x \bullet ((x' \bullet y') \bullet z') \rightarrow ((x \bullet x') \bullet y') \bullet z'$$

and from the superposition $(x \bullet ((x' \bullet y') \bullet z')$ we obtain

$$x \bullet (x' \bullet (y' \bullet z')) \rightarrow (x \bullet (x' \bullet y')) \bullet z'.$$

These rules then lead to further rules, and $OC(\underline{R})$ is infinite.

iv. Let $\underline{R} = \{f(x) \rightarrow g(x), g(h(x)) \rightarrow f(h(x))\}$. Then $OC(\underline{R})$ consists of R and the reflexive rules $f(h(x)) \rightarrow f(h(x))$ and $g(h(x)) \rightarrow g(h(x))$.

The overlap closure $OC(\underline{R})$ has a rich structure since the overlap closure construction preserves some properties of a rewriting system $\underline{R}$. The following theorem shows that every derived pair of two rewrite rules is also a rewrite rule, implying that the overlap closure $OC(\underline{R})$ is a rewriting system.

2.1. Theorem. If r,s,t,u are terms such that (r,s) and (t,u) are rewrite rules, then every derived pair $\langle p,q\rangle$ of (r,s) and (t,u) is also a rewrite rule.

Proof. One just has to verify that for each case in the definition of derived pair that every variable that occurs in q occurs also in p. $\square$

Let us consider some other properties, based on the properties of its rules, of a rewriting system $\underline{R}$

A term is said to be linear if no variable occurs in it more than once. A rewrite rule is left-linear if its left term is linear, right-linear if its right term is linear, and linear if its left and right terms are linear.

A rewriting system is called left-linear, right-linear, or linear, based on whether each of its rules is left-linear, right-linear, or linear, respectively. The following theorem implies that the overlap closure $OC(\underline{R})$ of a right-linear (left-linear, linear) $\underline{R}$ is also right-linear (left-linear, linear).

2.2. Theorem. If $r \to s$ and $t \to u$ are two right linear rules with disjoint variable sets, then each of their derived pairs, $\langle p, q\rangle$ is also right linear.

Proof. There are three cases:

(i) s unifies with the subterm t/i of t by their m.g.u. θ.

The corresponding derived pair $\langle p, q\rangle$ has

$$p = [\theta(t) \text{ with } \theta(r) \text{ at } i]$$

$$g = \theta(u)$$

Since s is linear, by Lemma 1 in the Appendix, substitutions for any two distinct variables in t/i in θ do not have a common variable. The variables in t other than the ones in t/i do not play any role. So $\theta(u)$ is linear.

(ii) the subterm s/i of s unifies with t by their m.g.u. θ.

The corresponding derived pair $\langle p, q\rangle$ has

$$p = \theta(r)$$

$q = [\theta(s) \text{ with } \theta(u) \text{ at } i]$

Since s/i is linear, by Lemma 1 in the Appendix, substitutions for any two distinct variables in t in θ do not have a common variable. So, $\theta(s)$ and $\theta(u)$ are linear, and q is thus linear.

(iii) if subterms of s do not unify with t, or s does not unify with subterms of t, then there are no derived pairs of $r \rightarrow s$ and $t \rightarrow u$. □

By a similar argument, it can also be proved that every derived pair of two left linear rules is left linear.

The name "overlap closure" comes from the fact that the rules of OC(**R**) are a subset of the transitive closure of the rewriting relation of **R**:

2.3. Lemma. If $p \rightarrow q$ is in OC(**R**) then $p \rightarrow^{+} q$ (using **R**).

Proof. By induction on the construction of $p \rightarrow q$ in OC(**R**). The basis of the induction is the case that $p \rightarrow q$ is included in OC(**R**) by virtue of being a rule of **R**. Then obviously $p \rightarrow^{+} q$ holds. If $(p \rightarrow q)$ is included in OC(**R**) by being a derived pair of (r,s) and (t,u) then by the induction hypothesis for the two rules (r,s) and (t,u), we have $r \rightarrow^{+} s$ and $t \rightarrow^{+} u$. By the definition of derived pair and the transitivity of $\rightarrow^{+}$, we then have $p \rightarrow^{+} q$. □

2.4. Corollary. If OC(**R**) contains a reflexive rule, $t \rightarrow t$, then the rewriting relation of **R** has a cycle.

Proof. Immediate from the above lemma. □

We would like to have the converse of this corollary, that if the rewriting relation of **R** has a cycle, then OC(**R**) contains a reflexive rule. This would permit searching for cycles by incrementally computing OC(**R**), looking for a reflexive rule. While we have not been able to prove this in full generality, we will present in the next section a restricted version and its proof. The proof is not easy, because the overlap closure of **R** is in general much smaller than the full transitive closure of **R**. It is this small size, relative to the transitive closure, however, that makes it feasible to use the overlap closure as the basis of an approach to proving uniform termination or, at least, a useful notion of "restricted termination," discussed in [Guttag, Kapur, and Musser, 1981].

3. REWRITE DOMINOES AND THE MAIN OVERLAP CLOSURE THEOREM

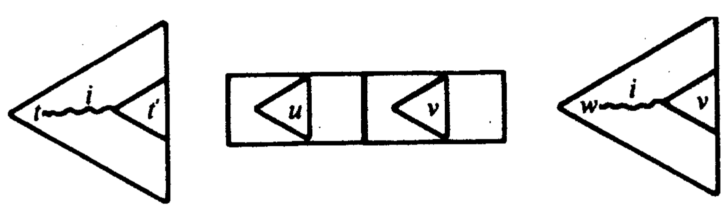
In order to be able to prove the major result about the overlap closure, we need to be able to deal precisely with the various cases of overlap between successive applications of rewrite rules in a rewrite sequence. We have found it useful to introduce a new representation of rewriting that helps to make such cases clear.

The domino representation (or rewrite domino) of a rewrite rule is a rectangle divided into left and right halves in which are inscribed tree representations of the left and right terms of the rule. Function symbols in the terms are represented by labelled circles in the trees. Variable symbols are represented by labeled rectangles, called "variable boxes." For examples of some rules and their corresponding rewrite dominoes, see Figure 1.

For each kind of domino (that is, each domino corresponding to a specific rule), we assume there is an infinite stock of dominoes of that kind with their variable rectangles filled in with all possible terms. For each such domino, we also assume an infinite number of copies are available in the stock.

A sequence of rewrites can be represented by a domino layout, which is a two-dimensional arrangement of dominoes that obeys the rules of matching corresponding to those of term rewriting (Section 2). Before giving the formal definition of a layout, we refer the reader to an example of a rewrite sequence using the rules given in Figure 1 and its corresponding domino layout as shown in Figure 2. Another example is in Figure 3, and the two layouts in Figures 2 and 3 could be concatenated to give a single longer layout.

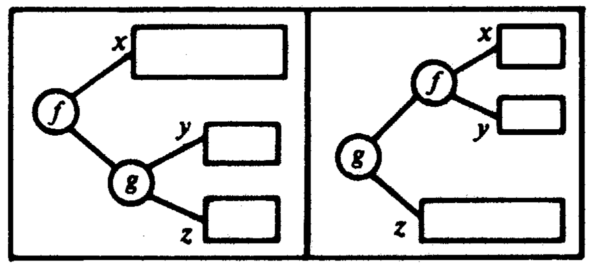
We draw trees oriented sideways with the root at the left, and we will use nested triangles to represent trees schematically. We define a unit layout from t to w to be a horizontal arrangement of a tree t, a domino with trees u and v, and another tree w,



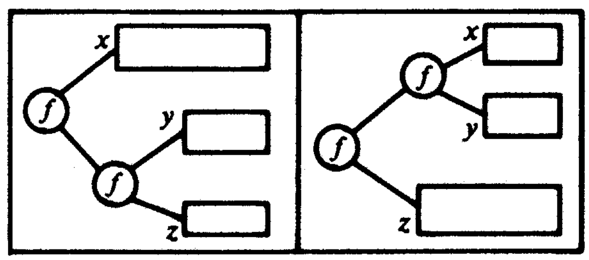
in which

RULE	DOMINO

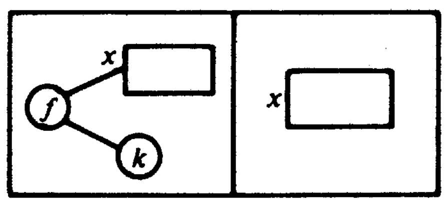
1. $f(x,g(y,z)) \rightarrow g(f(x,y),z)$



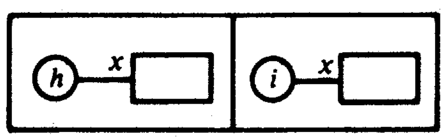
2. $f(x,f(y,z)) \rightarrow f(f(x,y),z)$



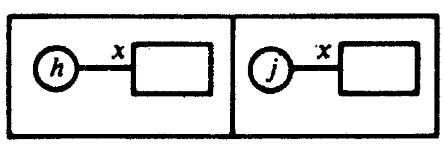
3. $f(x,k()) \rightarrow x$



4. $h(x) \rightarrow i(x)$



5. $h(x) \rightarrow j(x)$



6. $f(i(x),j(x)) \rightarrow l(x)$

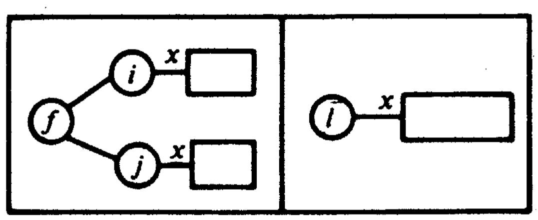
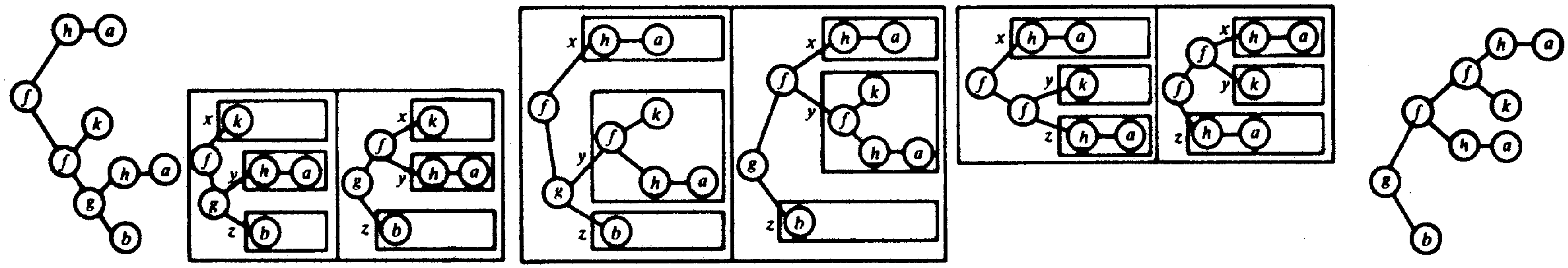
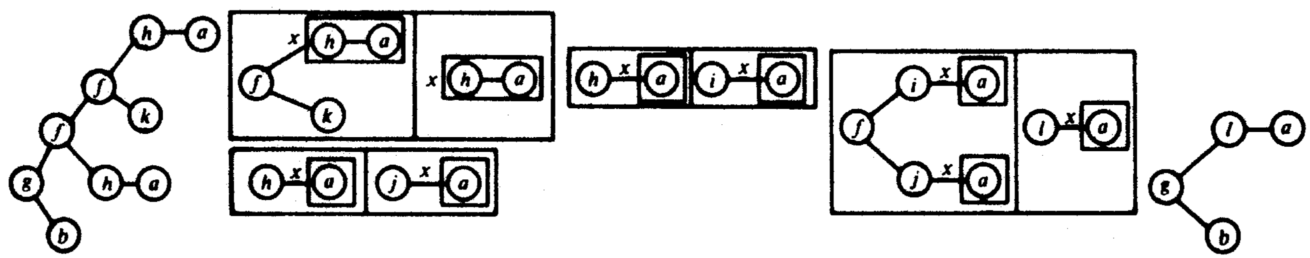


Figure 1. A set of rewrite rules and their corresponding rewrite dominoes.



$f(h(a),f(k,g(h(a),b))) \xrightarrow{1} f(h(a),g(f(k),h(a),b)) \xrightarrow{1} g(f(h(a),f(k,h(a))),b) \xrightarrow{2} g(f(f(h(a),k),h(a)),b)$

Figure 2. **A rewrite domino layout and the corresponding rewriting sequence (using dominoes of Figure 1).**



$g(f(f(h(a),k),h(a)),b) \xrightarrow{3,5} g(f(h(a),j(a)),b) \xrightarrow{4} g(f(i(a),j(a)),b) \xrightarrow{6} g(l(a),b)$

Figure 3. **Another layout (a continuation of the layout in Figure 2).**

1. at some position, i, in t there is a subtree t' that is identical to u, ignoring the variable boxes that appear in u;
2. the roots of t' and u are horizontally aligned;
3. w is the tree [t with v at i] and the roots of t and w are horizontally aligned.

A layout from t to v is defined as

1. a unit layout from t to v; or
2. the concatenation of a layout from t to u with a layout from u to v, with both copies of u dropped from the arrangement; or
3. any arrangement obtained from a layout by translating horizontally any domino, as long as no other domino or end tree is overlaid or crossed (this allows compaction of a layout by placing one domino above another when they match disjoint subterms).

The examples in Figures 2 and 3 illustrate a number of observations we can make about this representation of rewriting:

1. In a domino layout there is no distinction between different orders of rewriting when the rules are being applied to disjoint subterms; e.g., the layout in Figure 3 would not be different if rule 5 had been applied before rule 4 or before rule 3. One can think of these rules being applied in parallel, since the order of application is always immaterial in this case. The layout representation just makes this property especially evident.
2. To the property that "the rightmost term of a rewrite sequence is terminal" corresponds the property that "there is no way to play a domino on the layout" (formally, there is no way to concatenate a unit layout onto the layout). The layout is said to be blocked. (The layout in Figure 3 is blocked.)
3. Thus the rules have the uniform termination property if and only if every possible layout eventually is blocked. Equivalently, there are no infinite layouts.

Our purpose with this representation of rewriting is to provide a conceptual tool for finding and presenting proofs of new results about term rewriting systems. The first result we will prove with the aid of rewrite dominoes is one that will allow us to speed up the search for cycles by considering only those sequences of rewrites in which a "major rewrite" occurs.

A rewrite $t_0 \to t_1$ is called a major rewrite if it is by application of a rule, $t \to u$, to the entire term t_0; i.e., for some substitution θ, $\theta(t) = t_0$ and $\theta(u) = t_1$. When only a proper subterm of t_0 is matched, $t_0 \to t_1$ is called a minor rewrite.

In a layout, a domino is called a <u>major domino</u> (of the layout) if it represents a major rewrite, and a <u>minor domino</u> otherwise. Pictorially, major dominoes are those that span the width of the layout.

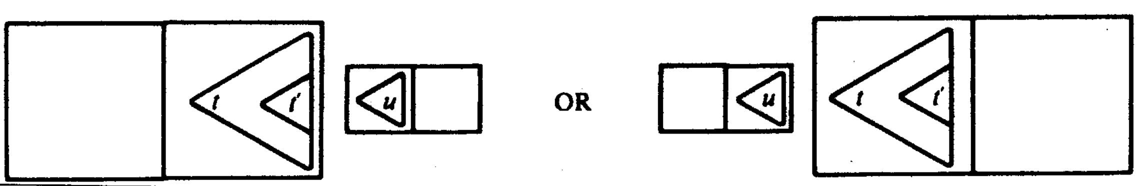
A <u>major cycle</u> is a cycle in which at least one of the rewrites is major.

3.1. <u>Theorem</u>. If a rewriting relation has a cycle, it has a major cycle.

<u>Proof</u>. Let us define the <u>corridor</u> of a domino in a layout to be the horizontal strip across the layout determined by the position and width of the domino:

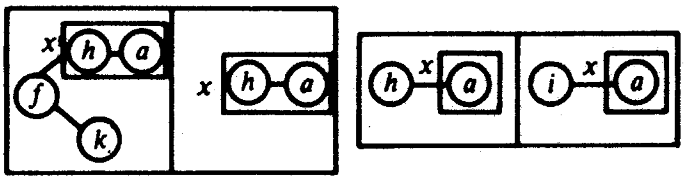
Any two corridors in a layout are either disjoint or one is contained in the other. Therefore, we can find a corridor that is spanned by a domino and which contains a layout as follows: start with any leftmost domino and follow its corridor to the right; whenever a domino is encountered that doesn't lie in the corridor, adopt its corridor. When we reach the right end, we have a corridor containing a layout including a domino that is major with respect to it. If the whole layout is cyclic, the identified layout will be also, and will represent a major cycle. □

We now want to define some terminology and some manipulations of layouts that will be useful in proving theorems about the overlap closure of a set of rules. Consider an adjacent pair of dominoes in a layout. Let t and u be the trees on the adjacent halves, where a subtree t' of t is identical to u (possibly t' = t):

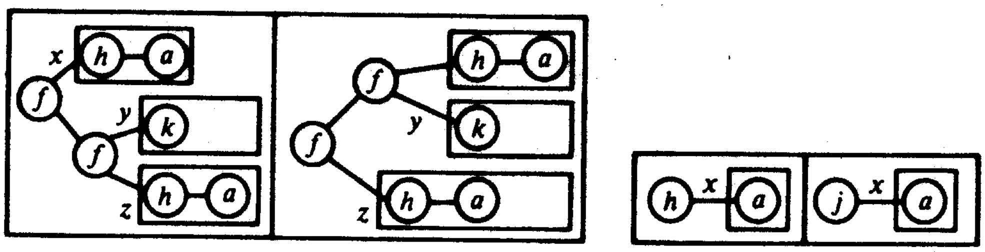


If either of t' or u is contained entirely within a variable box, i.e., the match is not between two nonvariable subterms, we say that the pair of dominoes is __weakly__ __matched__, and otherwise that it is __strongly__ __matched__.

__Examples__. In Figure 3, the domino pair



is weakly matched. Similarly the pair



that appears in the concatenation of the layouts of Figures 2 and 3 is weakly matched, while all the other adjacent pairs are strongly matched.

Now suppose we have two weakly matched dominoes, as in Figure 4a, where t′ is contained in the x variable box. If the (s,t) domino is right-linear (i.e., t is linear), then the pair of dominoes can be __transposed__ as follows: remove the (u,v) domino from the layout and move the (s,t) domino to the right, so that copies of the (u,v) domino can be inserted to the left of the (s,t) domino, one adjacent to each x box in s (see Figure 4b). Then __the resulting configuration is still a layout__, (the dominoes all match, using the same set of rules) with the same end trees. This is the case also when a symmetric kind of transposition is performed on the layout in Figure 5a, producing the layout in Figure 5b, where we assume that the (u,v) domino is left-linear.

Such transpositions cannot necessarily be performed on strongly matched dominoes, but we will define a different kind of manipulation for this case. Strong matching corresponds to the concept of overlapping in the definition of derived pairs: if (r,s) and (t,u) are rules that have a derived pair ⟨p,q⟩, then the dominoes corresponding to (r,s) and (t,u) can be placed in a layout so that they are strongly matched. The layout configuration shows just where the strong match occurs and identifies a potential derived pair.

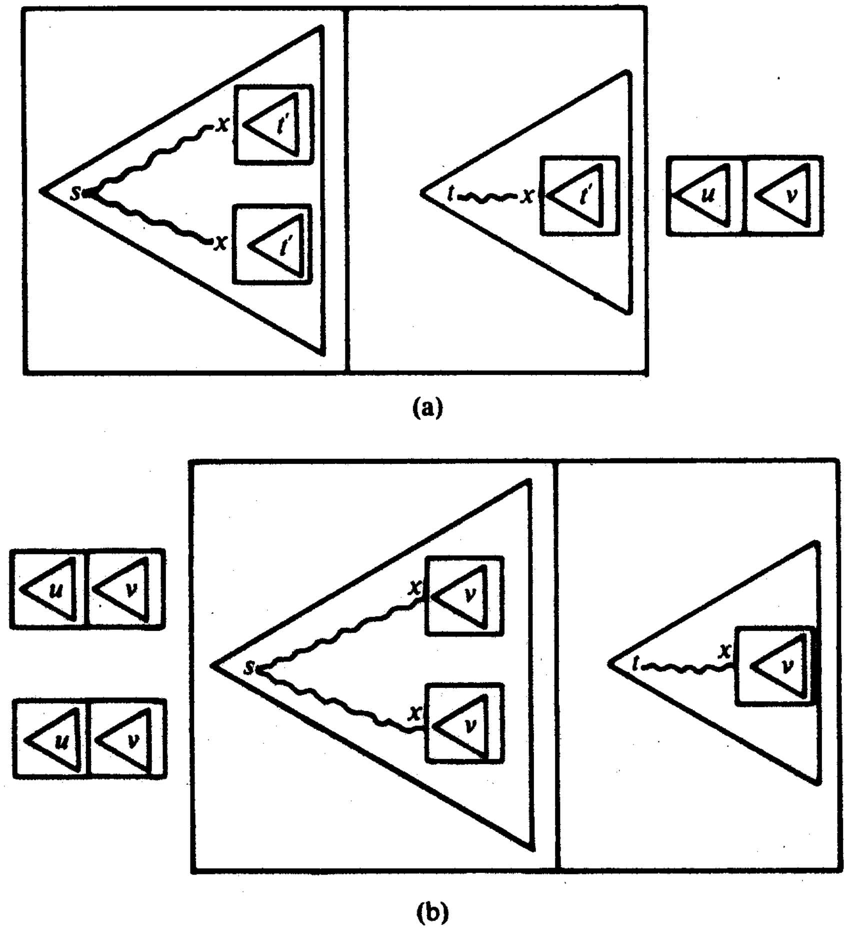


Figure 4. Transposition of weakly matched dominoes, where left domino is right-linear.

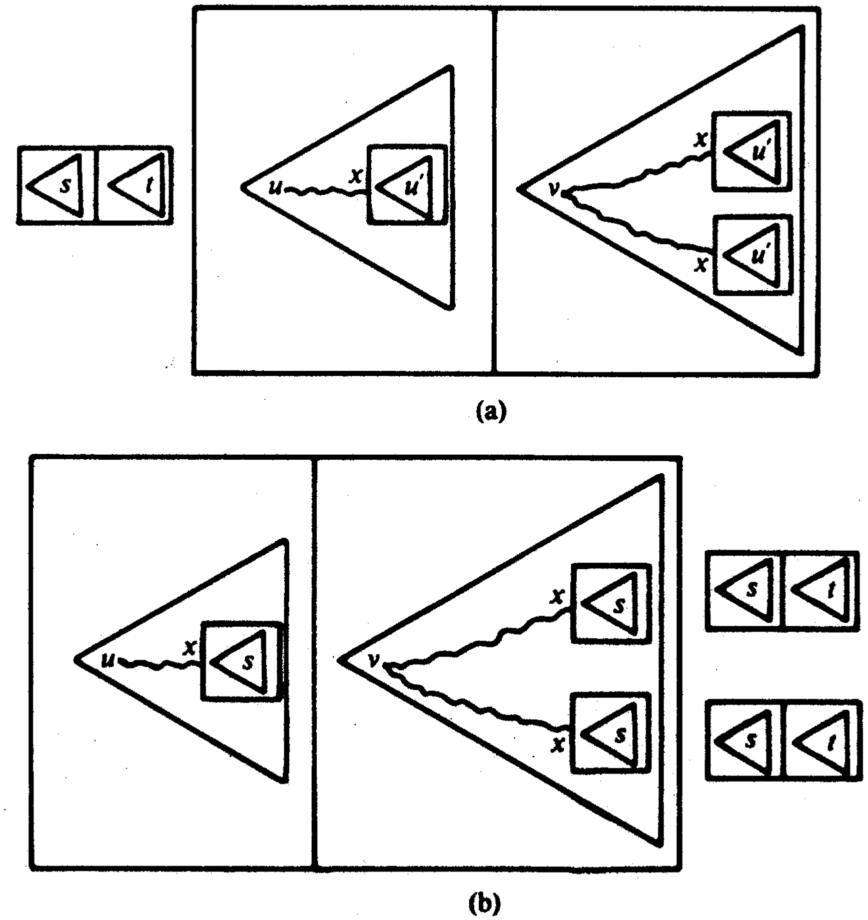


Figure 5. Transposition of weakly matched dominoes, where right domino is left-linear.

Suppose now that instead of our stock of dominoes corresponding to a given rule set **R**, we have a stock corresponding to OC(**R**), the overlap closure of **R**. Then for any strongly matched pair of dominoes in a layout there is a domino in our stock which corresponds to a derived pair generated by the matching pair. By Lemma 3 proved in the Appendix, we can replace the strongly matched pair in the layout by the "derived pair domino" thus identified, and the result will still be a layout with the same end trees.

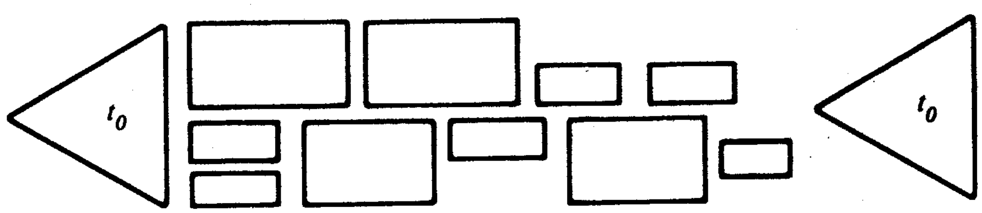
We are now in a position to prove:

3.2. Theorem. Suppose the rewriting relation of **R** is globally finite and every rule in **R** is right-linear. If the rewriting relation of **R** has a cycle, OC(**R**) contains a reflexive rule.

Proof. (By construction.) Let

$$(*) \qquad t_0 \to t_1 \to \cdots \to t_n \to t_0$$

be a given cycle. Corresponding to (*) is a cyclic domino layout

(**) 

where the dominoes correspond to rules of R. In fact since each of these rules is also in OC(R), we may take this layout as a layout of dominoes corresponding to rules of OC(R). We will show how to manipulate this layout to a form that shows there is a reflexive rule $t \to t$ in OC(R).

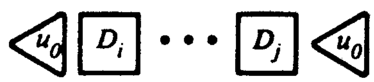
We describe the manipulations as an algorithm operating on the cyclic layout (**).

Step 1. [Extract major cycle.] As in the proof of Theorem 5.1, extract from (**) a sublayout representing a major cycle, making it the layout subject to the following steps. Also replace t_0 with its subterm matched by the layout.

Step 2. [Push major dominos to right end.] Manipulate the layout to a form in which all of the major dominoes are together at the right end, by means of transpositions or replacements by derived pair dominoes: whenever **D** is a major domino and **E** is a minor domino adjacent to **D** on the right either $\underline{D}$ and $\underline{E}$ are weakly matched, in which case they can be transposed, or they are strongly matched, in which case they can be replaced by the derived pair domino they define - which is a major domino. This derived pair domino is also right linear, as Lemma 1 in the Appendix shows.

Step 3. [Look for cycle among major dominoes.] There is now a nonempty sequence of major dominoes $D_1, \ldots, D_m$ at the right end of the layout:

These dominoes can only be strongly matched - except for the case where the right-hand side of D_i is just a variable, but shortly we will show that such a possibility can be ruled out. If there is some contiguous subsequence $D_i, \ldots D_j$ that forms a cyclic layout



then, since there can only be strong matches, these dominoes can be combined by $j - i + 1$ replacements into a single domino $\underline{D}$ that forms a cyclic layout:

Let $\underline{D}$ represent (p,q). Then there is a substitution θ such that $u_0 = \theta(p)$ and $\theta(q) = u_0$, i.e., θ unifies p and q. Furthermore, a derived pair of (p,q) and (p,q) is the reflexive rule $(\theta(p), \theta(q))$. Since this is in OC($\underline{R}$), we terminate the algorithm.

Step 4. [Duplicate.] If no such subsequence exists, construct a copy of the layout adjacent to it and return to Step 2 with the resulting layout:

That concludes the statement of the algorithm. Before considering the question of termination of the algorithm, we dispense with the detail mentioned in Step 3: the case of adjacent major dominoes $\underline{D}$ and $\underline{E}$ where the right term u of $\underline{D}$

is just a variable. We can assume the left term t of $\underline{D}$ is not just a variable (if it were then it would have to be the same variable as u and we would already have a reflexive rule). Since the layout is cyclic, if we drop $\underline{D}$ from the layout, we obtain a layout that has as its right end term a proper subterm identical to the left end term. From this we conclude that the term rewriting relation is not globally finite, contrary to assumption. This contradiction rules out the case under discussion.

It is obvious that each step of this algorithm is effective and terminating. Overall termination is guaranteed by the following facts:

a. At the k th execution of Step 2, the number of major dominoes, m, at the right end is at least 2^k.

b. Let $t_0'[k]$ denote the term to the left of D_1 in the layout at the k th execution of Step 3. Since each $t_0'[k]$ is derived from t_0 and the rewriting relation is globally finite, there are only finitely many distinct possibilities for $t_0'[k]$. By a), then, there is one such term for which arbitrarily long layouts of major dominoes exist. Again by global finiteness, these layouts cannot all continue without producing a term, u_0, that is a duplicate of some term previously obtained in the layout.

Since the algorithm always terminates, and does so with a reflexive rule in OC($\underline{R}$), this proves the theorem. □

The corresponding theorem obtained by replacing "right-linear" by "left-linear" can also be proved in a similar manner. Combining these theorems with Corollary 4.3, we have:

3.3. <u>Theorem</u>. Suppose the rewriting relation of $\underline{R}$ is globally finite and every rule in $\underline{R}$ is right-linear or every rule in $\underline{R}$ is left-linear. Then the rewriting relation of $\underline{R}$ is uniformly terminating if and only if OC($\underline{R}$) contains no reflexive rule.

Some applications of this theorem are explored in [Guttag, Kapur, and Musser, 1981]

Recently, Dershowitz (1981) has proposed a "forward chain" construction for rewriting systems and proved that a right-linear rewriting system is uniformly terminating if and only if it has no infinite forward chains. However, for left-linear systems the analogous result requires that the left-hand sides of the rules be nonoverlapping, a problem that we had independently encountered when considering the forward chain construction and a similar backward chain construction. We were thus led to invent the overlap closure construction. The following example from Dershowitz (1981) illustrates the advantage of the overlap closure construction over forward chains. Using the forward chain construction, it is not possible to determine the nontermination of this left-linear rewrite system, as pointed out by Dershowitz. The rewriting system is

$$f(a(),\ b(),\ x) \rightarrow f(x,\ x,\ b())$$

$$b() \rightarrow a().$$

These rules have only two forward chains, both finite:

$$f(a(),\ b(), x) \Rightarrow f(x,\ x,\ b()) \Rightarrow f(x,\ x,\ a()), \text{ and } b(\ ) \Rightarrow a(),$$

but we cannot conclude anything about the termination of the rules because they are not right-linear and, although they are left-linear, the left-hand sides are overlapping. But in the overlap closure construction, the rules have a derived pair rule

$$f(b(),\ b(),\ x) \rightarrow f(x,\ x,\ b()),$$

which, when overlapped with itself, gives the reflexive rule

$$f(b(),\ b(),\ b()) \rightarrow f(b(),\ b(),\ b()\ )),$$

as a derived pair, proving that the rules are nonterminating.

4. CONCLUSION

We have discussed two ways to make use of finite subsets of the overlap closure: proving restricted termination and disproving uniform termination. We have explored, without much success, using such finite subsets as parts of proofs of uniform termination. We conjecture that for certain classes of term rewriting systems it should be possible to compute a bound, n, such that if a cycle exists, there exists a cycle in which every term is of size n or less. For such classes, the overlap closure would provide a decision procedure for uniform termination.

Another open question about the generality of the overlap closure construction is whether the assumption of left-linearity or right-linearity is necessary. Although we have not been able to find proofs of our results without this assumption, we have also been unable to construct a counterexample. In any case, as discussed above, the overlap closure construction is more general than either forward or backward chain constructions.

For the class of term rewriting systems to which it may be applied, constructing the overlap closure is as useful as constructing the complete transitive closure. Furthermore, using the overlap closure to show restricted termination or the absence of uniform termination will always involve computing fewer terms than would using the transitive closure. We do not yet have much empirical or analytical evidence as to the absolute efficiency of using the overlap closure for these purposes. The key question is how many terms must be examined in order to demonstrate that no cycle is possible for terms of up to size n. The few examples we have tried, using a preliminary implementation, we have found encouraging.

ACKNOWLEDGMENTS

The basic idea of conducting a search for repeated terms (cycles) or subterms sprang from discussions in 1977 between one of the authors (Musser) and Dallas Lankford. We thank P. Gloess, G. Huet, and J. Levy for their interest and assistance in refuting some of our earlier conjectures, thus helping us arrive at the notion of the overlap closure and the theorems of Section 3. We also thank P. Narendran for assistance in constructing the proof of the theorems in the Appendix and J. Goguen for discussions of the approach to term rewriting used in OBJ. John Guttag's research is supported in part by the National Science Foundation under grant MCS78-01798 and by an Office of Naval Research Contract with DARPA funding #N00014-75-C-0661.

REFERENCES

N. Dershowitz, ''Termination of Linear Rewriting Systems - Preliminary Version,'' in Automata, Languages, and Programming, Eighth Colloquium, Israel (Eds: S. Even and O. Kariv), LNCS 115, Springer Verlag, New York, 1981.

J.A. Goguen and J. Tardo, ''An Introduction to OBJ-T,'' Proc. of Conf. on Specification of Reliable Software (1979).

J.V. Guttag, E. Horowitz, and D.R. Musser, ''Abstract Data Types and Software Validation,'' Comm. A.C.M. 21 (1978), 1048-1064.

J.V. Guttag, D. Kapur, and D.R. Musser, On Proving Uniform Termination and Restricted Termination of Rewriting Systems, Report No. 81CRD272, G.E. C. Research and Development, Schenectady, New York, Nov. 1981.

G. Huet, ''Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems,'' J. ACM 27 (1980), 797-821.

G. Huet, D.S. Lankford, ''On the Uniform Halting Problem for Term Rewriting Systems,'' Rapport Laboria 283, IRIA, Mars 1978.

G. Huet, D.C. Oppen, ''Equations and Rewrite Rules: a Survey,'' Formal Languages Perspectives and Open Problems (R. Book, ed.), Academic Press (1980).

D.E. Knuth, P. Bendix, ''Simple Word Problems in Universal Algebra.'' Computational Problems in Abstract Algebra (J. Leech, ed.), Pergamon Press (1970), 263-297.

D. Musser, ''Abstract Data Type Specification in the AFFIRM system.'' IEEE Transactions on Software Engineering 6 (1980), 24-31.

G.E. Peterson and M.E. Stickel, Complete Sets of Reductions for Equational Theories With Complete Unification Algorithms, Tech. Report, Dept. of Computer Science, U. of Arizona, Tucson, Sept. 1977.

APPENDIX

1. <u>Lemma</u>. Let t and u be unifiable terms with disjoint variable sets, and θ be their most general unifier. Let θ^* be the restriction of θ to the variables of u, say $\theta^* = [e_1/v_1,\ldots,e_n/v_n]$. If t is linear, then all variables in $e_1,\ldots,e_n$ are distinct.

<u>Proof</u>. For every variable x having k (>1) occurrences in u, replace different occurrences of x by distinct variables $x_1,\ldots,x_k$ that do not appear in t and u. Let u' be the resulting term, which is linear.

By Lemma 2, in the m.g.u. θ' of t and u', substitutions for distinct variables in t and u' do not have a common variable. Let σ_x be the m.g.u. for the set of terms $\theta'(x_i)$, $1 \le i \le k$, the substitutions for the variables used to replace multiple occurrences of x in u. If these σ_x for every variable x having multiple occurrences in u are composed with θ', we get a unifier of t and u.

In this unifier, substitutions for variables in u do not have a common variable. From this, it is evident that the m.g.u. θ of t and u cannot have substitutions for variables in u that share common variables. □

2. <u>Lemma</u>. For two unifiable terms t and u, if t and u are linear, then the substitutions in their m.g.u. θ for any two distinct variables of t or u do not have common variables.

<u>Proof</u>. By induction on the structure in term t.

Basis: t is a variable.

Then $\theta(t) = u$ and the statement trivially holds.

Inductive step: $t \equiv f(t_1,\ldots,t_n)$

For t and u to be unifiable, either u is a variable or $u == f(u_1,\dots,u_n)$. The case of u being a variable is handled as in the basis step.

For the case $u == f(u_1,\dots,u_n)$, for each i, $1\le i\le n$, t_i must unify with u_i by their m.g.u. θ_i, say. By the inductive hypothesis, the statement holds for each of θ_i. Since t and u are linear, the disjoint union of θ_i, $1\le i\le n$, is the m.g.u. θ of t and u. It follows that the statement of the lemma holds for θ also. □

3. __Lemma__. Suppose $t_0 \rightarrow t_1$ using $r \rightarrow s$ applied at position i, $t_1 \rightarrow t_2$ using $t \rightarrow u$ applied at i.j, and s/j and t overlap determining the derived pair $\langle p,q\rangle = \langle \theta(r),[\theta(s)$ with $\theta(u)$at $j]\rangle$. Then $t_0 \rightarrow t_2$ using $p \rightarrow q$ applied at i. A similar result holds for the case in which s unifies with a subterm of t.

__Proof__. Rename the variables of t and u, if necessary, so that s and t have no variable in common. There is some subterm t_0/i and a substitution θ_1 such that $\theta_1(r) = t_0/i$ and $t_1 = [t_0$ with $\theta_1(s)$ at i].

Again, there is some subterm $t_1/(i.j)$ and a substitution θ_2 such that $\theta_2(t) = t_1/(i.j)$ and $t_2 = [t_1$ with θ_2 (u) at i.j].

Since the variables of s and t are disjoint, we have $(\theta_1 \cup \theta_2)(s/j) = \theta_1(s/j) = \theta_2(t) = (\theta_1 \cup \theta_2)(t)$. That is, $\theta_1 \cup \theta_2$ is a unifier of s/j and t and therefore has θ as a factor:

$$\theta_1 \cup \theta_2 = \theta_3 \bullet \theta, \text{ for some substitution } \theta_3.$$

Thus $t_0/i = \theta_1(r) = (\theta_1 \cup \theta_2)(r) = (\theta_3 \bullet \theta)(r) = \theta_3(\theta(r)) = \theta_3(p)$. That is, t_0 is matched by p at i. Now consider $\theta_3(q)$; it is

$$\theta_3\ ([\theta(s) \text{ with } \theta(u) \text{ at } j])$$
$$= [\theta_3(\theta(s)) \text{ with } \theta_3(\theta(u)) \text{ at } j]$$

$$= [\theta_1(s) \text{ with } \theta_2(u) \text{ at } j].$$

Thus $t_2 = [t_1 \text{ with } \theta_2(u) \text{ at } i.j]$

$$= [[t_0 \text{ with } \theta_1(s) \text{ at } i] \text{ with } \theta_2(u) \text{ at } i.j]$$

$$= [t_0 \text{ with } [\theta_1(s) \text{ with } \theta_2(u) \text{ at } j] \text{ at } i]$$

$= [t_0 \text{ with } \theta_3(q) \text{ at } i]$, showing that

$t_0 \rightarrow t_2$ using $p \rightarrow q$ applied at i. We omit the proof of the case in which s unifies with a subterm of t. □

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: MIT/LCS/TR-268	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: "Derived Pairs, Overlap Closures, And Rewrite Dominoes: New Tools For Analyzing Term Rewriting Systems"	5. TYPE OF REPORT & PERIOD COVERED: Technical Report/Dec. '81
	6. PERFORMING ORG. REPORT NUMBER: MIT/LCS/TR-268
7. AUTHOR*(s)*: John V. Guttag, Deepak Kapur, David R. Musser	8. CONTRACT OR GRANT NUMBER*(s)*: MCS78-01798, N00014-75-C-0661
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, Ma. 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: DARPA, 1400 Wilson Blvd., Arlington, Va. 22209	12. REPORT DATE: December 1981
	13. NUMBER OF PAGES: 19
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*: Office of Naval Research, Department of the Navy, Information Systems Program, Arlington, Va. 22217	15. SECURITY CLASS. *(of this report)*: Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

This document is approved for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Confluence, Critical pair, Derived pair, Knuth-Bendix, Overlap closure, Resricted termination, Rewrite dominoes, Term Rewriting Systems, Uniform termination.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Starting from the seminal work of Knuth and Bendix, we develop several notions useful in the study of term rewriting systems. In particular we introduce the notions of "derived pairs" and "overlap closure" and show that they are useful in analyzing sets of rewrite rules for various properties related to termination. We also introduce a new representation, based on rewrite dominoes, for rewrite rules and sequences of rewrites.