Coloured rippling: An extension of a theorem proving heuristic

Tetsuya Yoshida[1] Alan Bundy[1] Ian Green[1] Toby Walsh[2] David Basin[3]

Abstract. *Rippling* is a type of rewriting developed in inductive theorem proving for removing differences between terms; the induction conclusion is annotated to mark its differences from the induction hypothesis and rippling attempts to move these differences. Until now rippling has been primarily employed in proofs where there is a single induction hypothesis. This paper describes an extension to rippling to deal with theorems with multiple hypotheses. Such theorems arise, for instance, when reasoning about data-structures like trees with multiple recursive arguments. The essential idea is to colour the annotation, with each colour corresponding to a different hypothesis. The annotation of rewrite rules used in rippling is similarly generalized so that rules propagate colours through terms. This annotation guides search so that rewrite rules are only applied if they reduce the differences between the conclusion and some of the hypotheses. We have tested this implementation on a number of problems, including two of Bledsoe's challenge limit theorems.

1 INTRODUCTION

Rippling is a powerful heuristic for inductive theorem proving which tries to remove the differences between terms [4]. The induction conclusion is annotated to mark its differences from the induction hypothesis. Rippling moves these differences using annotated rewrite rules. The annotation can be thought of dividing the term tree in the conclusion into two parts: the *skeleton* and the *wave-fronts*. The skeleton is an image of the induction hypothesis in the induction conclusion and this remains undisturbed during rippling. The wave-fronts represent those parts of the term that need to be moved "out of the way" by rippling. Rippling is an effective strategy since if, say, we can move all the wave-fronts to the top of the term tree, then we will be able to appeal to the induction hypothesis since an exact image of the hypothesis now appears in the conclusion. The rippling heuristic has been implemented in a "proof planning" system called CL^AM.

Until now, rippling has mainly been applied in domains where recursive data-types have only one recursive argument. Many theorems have been proven about data structures like numbers and lists (which are built from smaller numbers and smaller lists) but fewer about data-structures like trees which are built from *multiple* subtrees. The reason for this is that the annotation used by rippling provides an invariant (the skeleton) that directs the proof. When there is a single induction, this invariant corresponds to a single term and is captured unambiguously by the annotation used by rippling. However, with multiple recursive arguments there are multiple induction hypotheses and thus multiple invariants. For the structure of these multiple invariants to be simultaneously preserved during the proof, we must generalize how annotations are used to represent differences. We therefore propose that each hypothesis be associated with a different colour. Annotation in the induction conclusion is then coloured to distinguish between the occurrences of the hypotheses in the skeleton. The annotated rewrite rules used by rippling are extended to manipulate such coloured annotation. Such extensions are also useful for guiding non-inductive proofs which have multiple hypotheses.

Coloured rippling has been implemented in the CL^AM system. We describe this and the results of experiments which include the application of coloured rippling to solve two challenge theorems about limits.

In the first two sections we introduce rippling in more detail, and show why colours are needed to control rippling towards multiple hypotheses. In §4 we describe the implementation of coloured rippling and several experiments using it. In §5 we discuss related work. Finally we draw conclusions and indicate directions for further work.

2 RIPPLING

The recursive type $tree$ of (binary) trees has two constructors $leaf$ and $node$. We define functions $maxht$ and $minht$ as follows:

$$\begin{aligned} maxht(leaf(n)) &= 0 \\ maxht(node(t_1, t_2)) &= s(max(maxht(t_1), maxht(t_2))) \end{aligned}$$

(similarly for $minht$).

Consider a simple theorem, *maxht-minht*, about binary trees,

$$\forall t : tree(pnat).maxht(t) \geq minht(t)$$

which CL^AM proves via structural induction over binary trees (see [6] for how CL^AM chooses inductions), giving *two* hypotheses in the step case,[4]

$$\begin{aligned} maxht(l) &\geq minht(l), \\ maxht(r) &\geq minht(r) \end{aligned}$$

and an *annotated* induction conclusion

$$maxht(\boxed{node(\underline{l}, \underline{r})}^{\uparrow}) \geq minht(\boxed{node(\underline{l}, \underline{r})}^{\uparrow}) \qquad (1)$$

Wave-holes are those terms which are underlined; *wave-fronts* are boxed. All wave-fronts have at least one wave-hole. The idea is that the wave-fronts make explicit how the conclusion and hypotheses differ.

[1] Department of AI, University of Edinburgh, Edinburgh EH1 1HN, Scotland
[2] INRIA-Lorraine 615, rue du Jardin Botanique, 54602 Villers-les-Nancy, France
[3] Max-Planck-Institut für Informatik, Saarbrücken, Germany

[4] The base case is proved by symbolic evaluation.

ECAI 94. *11th European Conference on Artificial Intelligence* Edited by A. Cohn
Published in 1994 by John Wiley & Sons, Ltd.

A *skeleton* is defined as[5]

Definition 1 (skeleton) *We define $skel(t)$ for annotated terms t as*

$$skel(\boxed{f(t_1,\ldots,t_n)}) = \bigcup_i skel(t'_i) \quad \textit{for all } t_i = \underline{t'_i}$$
$$skel(f(t_1,\ldots,t_n)) = \{f(u_1,\ldots,u_n) \mid \forall i.u_i \in skel(t_i)\}$$
$$skel(a) = \{a\} \quad \textit{for atoms } a$$

The skeleton of a term is any element of $skel(t)$.

Terms with wave-fronts containing only a single wave-hole have a unique skeleton; those with multiple holes have multiple skeletons. We define $erase(t)$ to be t with all annotations removed.

Rippling is the process of moving the annotated differences between the conclusion and the hypotheses, whilst preserving (at least one) skeleton. Preserving the skeleton means that we are able to appeal to the induction hypothesis and complete the inductive proof should the annotation be moved to the top of the conclusion (or around positions which correspond to universal variables). This is done via the application of annotated rewrite rules, called *wave-rules* to the conclusion. Wave-rules are derived from equations, inequalities, equivalences and implications and written in the form $LHS \Rightarrow RHS$. CI^AM adds annotations so that

- $skel(LHS) \supseteq skel(RHS)$ (i.e., skeleton preserving);
- $|LHS| > |RHS|$ for some well-founded measure $|\cdot|$.

Rewriting with wave-rules is similar to normal term rewriting with the additional proviso that annotation in the rule must match with annotation in the conclusion. This severe restriction on the application of wave-rules greatly reduces search and the well-founded measure guarantees that rippling terminates, yet rules like associativity can be used in both directions (see [4]).

To ripple the induction conclusion (1) we can use the wave-rules shown in table 1 formed from the definitions of $maxht$, $minht$ and $\geq$.

$$maxht(\boxed{node(\underline{L},\underline{R})}^{\uparrow}) \Rightarrow \boxed{s(max(\underline{maxht(L)},\underline{maxht(R)}))}^{\uparrow}$$
$$minht(\boxed{node(\underline{L},\underline{R})}^{\uparrow}) \Rightarrow \boxed{s(min(\underline{minht(L)},\underline{minht(R)}))}^{\uparrow}$$
$$\boxed{s(\underline{X})}^{\uparrow} \geq \boxed{s(\underline{Y})}^{\uparrow} \Rightarrow X \geq Y$$

Table 1. Wave-rules for *maxht-minht* theorem

Rippling (1) with the wave-rules from table 1 gives

$$\boxed{max(\underline{maxht(l)},\underline{maxht(r)})}^{\uparrow} \geq \boxed{min(\underline{minht(l)},\underline{minht(r)})}^{\uparrow}$$

To continue rippling suppose we have

$$\boxed{max(\underline{U_1},\underline{U_2})}^{\uparrow} \geq \boxed{min(\underline{V_1},\underline{V_2})}^{\uparrow} \Rightarrow \boxed{\underline{U_1 \geq V_1} \wedge \underline{U_2 \geq V_2}}^{\uparrow} \quad (3)$$
$$\boxed{max(\underline{U_1},\underline{U_2})}^{\uparrow} \geq \boxed{min(\underline{V_1},\underline{V_2})}^{\uparrow} \Rightarrow \boxed{\underline{U_1 \geq V_2} \wedge \underline{U_2 \geq V_1}}^{\uparrow} \quad (4)$$

where rule (4) is a commuted version of rule (3). Now there is a choice point since both of these wave-rules are applicable; if wave-rule (4) is applied, the induction conclusion becomes

$$\boxed{\underline{maxht(l) \geq minht(r)} \wedge \underline{maxht(r) \geq minht(l)}}^{\uparrow}$$

which does not follow from the induction hypotheses. We incur a search penalty since we must apply (3) for the proof to go through. Note it is not fair to have (3) as a rewrite and not (4) since some proofs will need the latter.

3 COLOURED RIPPLING

Clearly the notion of skeleton preservation enforced on wave-rules (condition (ii) above) is not strong enough, since the *particular* skeletons being preserved are not present in the hypothesis. What is required is a way to distinguish between these multiple skeletons and so avoid 'mixing' one skeleton (corresponding to one hypothesis) with another (corresponding to another hypothesis).

3.1 Adding colours to wave-holes

The problem then is that multi-hole wave-rules permit the unwanted mixing of skeletons. Characterising when this happens is straightforward since CI^AM can identify each wave-hole with its corresponding induction hypothesis when it applies induction. We can therefore improve the utility of rippling by ensuring that *every* skeleton in the goal corresponds to an induction hypothesis. We do this with an explicit *colour* annotation on wave-holes which indicates the subterms of the induction conclusion corresponding to subterms of a particular induction hypothesis. Since a subterm may appear in more than one hypothesis, these annotations are in fact sets of colours.

We parameterize $skel$ with a colour, and ensure that $skel$ only collects holes of that colour (replace the first clause in definition 1 with:

$$skel_c(\boxed{f(t_1,\ldots,t_n)}) = \bigcup_i skel_c(t'_i) \quad \text{for all } t_i = \underline{t'_i}_S, c \in S$$

where S is the set of colours labelling a wave-hole).

$$maxht(\boxed{node(\underline{L}_A,\underline{R}_B)}^{\uparrow}) \Rightarrow \boxed{s(\max(\underline{maxht(L)}_A,\underline{maxht(R)}_B))}^{\uparrow}$$
$$minht(\boxed{node(\underline{L}_A,\underline{R}_B)}^{\uparrow}) \Rightarrow \boxed{s(\max(\underline{minht(L)}_A,\underline{minht(R)}_B))}^{\uparrow}$$
$$\boxed{s(\underline{X}_A)}^{\uparrow} \geq \boxed{s(\underline{Y}_A)}^{\uparrow} \Rightarrow X \geq Y$$

Table 2. Coloured wave-rules for *maxht-minht* (A, B stand for sets of colours)

Coloured rippling is a further restriction on monochromatic rippling: colours in the induction conclusion must also be matched with colour variables in the wave-rule. Coloured wave-rules are skeleton preserving *with respect to each colour*. Mixing of skeletons in the way described above is thus not possible (see §3.2).

Let us reconsider our example with coloured rippling. In the *maxht-minht* theorem the induction conclusion gets two colours (r and b say; we write the singleton colour set $\{c\}$ as c)

[5] We write $\boxed{f(\boxed{\underline{g(\underline{x})}})}$ as $\boxed{f(g(\underline{x}))}$ but all wave-fronts have holes immediately inside them.

$$maxht(\boxed{node(\underline{l}_b, \underline{r}_r)}^{\uparrow}) \geq minht(\boxed{node(\underline{l}_b, \underline{r}_r)}^{\uparrow}) \tag{5}$$

Equation (5) has two skeletons

$$skel_b = maxht(l) \geq minht(l)$$
$$skel_r = maxht(r) \geq minht(r)$$

not four as in the uncoloured case; note how these two skeletons are exactly the induction hypotheses.

Rippling (5) with the wave-rules from table 2 (the coloured varieties of those in table 1) gives,

$$\boxed{max(\underline{maxht(l)}_b, \underline{maxht(r)}_r)}^{\uparrow} \geq \boxed{min(\underline{minht(l)}_b, \underline{minht(r)}_r)}^{\uparrow} \tag{6}$$

Wave-rules (3) and (4) can only be coloured as,

$$\boxed{max(\underline{U_1}_A, \underline{U_2}_B)}^{\uparrow} \geq \boxed{min(\underline{V_1}_A, \underline{V_2}_B)}^{\uparrow} \Rightarrow \boxed{\underline{U_1 \geq V_1}_A \wedge \underline{U_2 \geq V_2}_B}^{\uparrow} \tag{7}$$

$$\boxed{max(\underline{U_1}_A, \underline{U_2}_B)}^{\uparrow} \geq \boxed{min(\underline{V_1}_B, \underline{V_2}_A)}^{\uparrow} \Rightarrow \boxed{\underline{U_1 \geq V_2}_A \wedge \underline{U_2 \geq V_1}_B}^{\uparrow} \tag{8}$$

As desired, wave-rule (8) will not match (6) since the colour annotations fail to match. Wave-rule (7) does match giving

$$\boxed{\underline{maxht(l) \geq minht(l)}_b \wedge \underline{maxht(r) \geq minht(r)}_r}$$

The proof is completed by appealing to the two induction hypotheses.

3.2 Creating coloured wave-rules

We have implemented an extension to CI^AM's existing (uncoloured) wave-rule parser which adds colour annotation to wave-holes. This is based on the notion of the *weakenings* of an annotated term.

Given a term t which has a wave-front with $k > 1$ wave-holes, we can *weaken* t by erasing up to $k - 1$ wave-holes. By erasing a wave-hole $\underline{t_i}$ we mean removing the underline annotation and erasing any further annotation in t_i. A term t is *maximally weak* (or *weakest*) when it cannot be further weakened. That is, every wave-front contains a single wave-hole. Let $weakenings(t)$ be the set of all maximal weakenings of t. For example, there are 4 maximal weakenings of the term in (1), i.e.,

$$\{\, maxht(\boxed{node(\underline{l}, r)}) \geq maxht(\boxed{node(\underline{l}, r)}),$$
$$maxht(\boxed{node(\underline{l}, r)}) \geq maxht(\boxed{node(l, \underline{r})}),$$
$$maxht(\boxed{node(l, \underline{r})}) \geq maxht(\boxed{node(\underline{l}, r)}),$$
$$maxht(\boxed{node(l, \underline{r})}) \geq maxht(\boxed{node(l, \underline{r})}) \,\}$$

The significance of this is that weakening reduces the number of skeletons and hence possibilities for skeleton mixing and we can use these maximally weak terms as the basis of an algorithm that assigns colours to wave-rules.

3.2.1 *Algorithm*

Accepting an uncoloured wave-rule, our parser creates weakened wave-rules with one wave-hole per wave-front and puts a different colour variable on each weakened wave-rule. Then these coloured wave-rules are combined into one wave-rule, which is identical with the original wave-rule except for colour variables on wave-holes.

The algorithm is as follows ($y := a$ means y takes the value a).

1. Receive the wave-rule constructed from the original parser as:

   $$LHS \Rightarrow RHS$$

   LHS and RHS are the left-hand side and right-hand side of the wave-rule.
2. $LWs := weakenings(LHS)$, $RWs := weakenings(RHS)$.
3. Put the *same* colour variable on the weakest with the *same* skeleton in *LWs* and *RWs*.
   (a) Create *LSWsList*, which is a list of lists. Its elements are the *weakest*s of LHS with the *same* skeleton. This process serves to sort and collect the *weakest*s with the *same* skeleton.
   (b) Create *RSWsList* in the same way as *LSWsList*.
   (c) For each element of *LSWsList*,
      i. *LSWs* := element of *LSWsList*
      ii. *RSWs* := element of *RSWsList* with the *same* skeleton as *LSWs*
      iii. Put the *same* colour variable on the *weakest*s in *LSWs* and *RSWs*
4. Combine the coloured *LSWs*s into a *NewLHS* and the coloured *RSWs*s into a *NewRHS*.
5. Output the coloured wave-rule as:

   $$NewLHS \Rightarrow NewRHS$$

3.3 Properties of coloured rippling

Coloured rippling is a restriction of the monochromatic case, i.e., fewer rule applications are allowed in the former. As monochromatic rippling terminates [4], coloured rippling must also terminate.

Since colours guide the construction of unmixed copies of the induction hypotheses, if coloured rippling succeeds on a conjecture, the proof can be completed using the fertilization tactic. This is not always true of monochromatic rippling. Coloured rippling therefore increases the 'utility' or expectancy of success of the rippling heuristic. Of course, there are (monochromatic) rippling proofs which are not admissible in the coloured case, but experiments in the context of inductive theorem proving suggests that this does not arise in practice (see the next section). In addition, since colour annotation reduces the applicability of wave-rules, search is even less of a problem in coloured rippling than in the monochromatic case.

4 EXPERIMENTS

This section describes some of the theorems proved using coloured rippling. (In fact, coloured rippling is able to prove all the theorems in the CI^AM corpus; see table 4 for a selection.)

4.1 Non-inductive Proofs

The CI^AM system was originally designed for inductive proof. However, rippling can also be used to construct proofs in any situation where lemmas, hypotheses, axioms *etc* are structurally similar to conjectures to be proved [7]. As an example of this, we show how coloured rippling has been used to prove limit theorems. These theorems were proposed by Bledsoe as benchmark theorems for automated theorem provers [2].[6] Limit theorems have multiple hypotheses; coloured rippling avoids mixing of these multiple hypotheses and thus uses less search than monochromatic rippling.

[6] Our thanks to Woody Bledsoe for suggesting the application of rippling to the LIM family of theorems.

Non-Inductive Scheme. In non-inductive proofs, it is necessary to create analogues of the induction hypotheses and conclusion by examining and annotating the conjecture to be proved. Such an examination is required during the proof-planning for LIM$^+$. The following method used in proving LIM+ is essentially CI^AM's existing normalization method augmented with a means of adding wave-front annotations to the conclusion.

Definition 2 (Non-inductive method) *The scheme for the non-inductive proof is defined as follows.*

1 *Assume that the conjecture is written in the following format*

$$t_1 \rightarrow t_2$$

2 *Convert the antecedent t_1 into conjunctive normal form (CNF) and treat the collection of disjuncts as hypotheses, $hyps$.*
3 *Treat the consequent t_2 as the conclusion.*
4 *Create the sequent:*

$$hyps \vdash t_2$$

5 *Add wave-front annotation to t_2 using difference matching (see §5) between t_2 and each hypothesis in $hyps$.*

The definition of lim is as follows

$$\lim(f, a, l) \equiv \forall\epsilon.(0 < \epsilon \rightarrow$$
$$\exists\delta.(0 < \delta \wedge \forall x.(x \neq a \wedge |x - a| < \delta \rightarrow |f(x) - l| < \epsilon)))$$

and the LIM$^+$ theorem is:

$$\lim(\lambda x.f_1(x), a, l_1) \wedge \lim(\lambda x.f_2(x), a, l_2) \rightarrow$$
$$\lim(\lambda x.f_1(x) + f_2(x), a, l_1 + l_2).$$

CI^AM begins the proof by choosing the non-inductive method. This results in a two-colour conclusion:

$$\lim(\lambda x.f_1(x), a, l_1), \lim(\lambda x.f_2(x), a, l_2) \vdash$$
$$\lim(\lambda x.\boxed{\underline{f_1(x)}_r + \underline{f_2(x)}_g}^{\uparrow}, a, \boxed{\underline{l_1}_r + \underline{l_2}_g}^{\uparrow})$$

There are no wave-rules applicable to this sequent. Motivated by a one-step lookahead which recognizes that rippling can then take place, CI^AM chooses to unpack the definitions of lim in the hypotheses and conclusion:

$$\forall\epsilon.(0 < \lfloor\epsilon\rfloor \rightarrow \exists\delta.(0 < \delta \wedge \forall x.(x \neq a \wedge |x - a| < \delta \rightarrow$$
$$\left|\boxed{(\underline{f_1(x)}_r + \underline{f_2(x)}_g)}^{\uparrow} - \boxed{(\underline{l_1}_r + \underline{l_2}_g)}^{\uparrow}\right| < \lfloor\epsilon\rfloor)))$$

(We have omitted the two hypotheses due to lack of space — bear in mind that each has been unpacked as well; we also drop the outermost quantifiers for f and l etc.) The $\lfloor\cdot\rfloor$ annotation marks positions in the conclusion which correspond to the positions of universal variables in both hypotheses. These positions are called *sinks* since they are able to absorb term structure resulting from rippling wave-fronts inwards [4].

Coloured rippling with the wave-rules from table 3, followed by two applications of fertilization with each of the hypotheses completes the proof. Most of the wave-rules from this table are derived from Bledsoe's clausal axiomatization. Those which are not are the distributivity of minus over plus (11),[7] and of implies over conjunction (12). Wave-rule (13) is an existential form of the lemma $U < X \wedge U < Y \rightarrow U < \min(X, Y)$, which appears in Bledsoe's axiomatization.

[7] Bledsoe's axiomatization instead uses unary minus with commutativity and associativity of plus.

4.2 Experimental Results

Coloured rippling has been incorporated into CI^AM, and successfully tested on a collection of inductive and non-inductive theorems. In the near future we hope to carry out some experimental analysis of the search behaviour. A selection of theorems about binary trees is provided below; we hope that quantitative results concerning efficiency will be forthcoming.

$maxht(t) \geq minht(t)$
$(*)\ tipcount(swap(t)) = tipcount(t)$
$tipcount(duptree(t)) > tipcount(t)$
$maxht(duptree(t)) > maxht(t)$
$(*)\ swap(swap(t)) = t$
$(*)\ flattentree(swap(t)) = rev(flattentree(t))$
$length(flattentree(t)) = tipcount(t)$
$flattentree(maptree(t, f)) = mapcar(flattentree(t), f)$
$swap(maptree(t, f)) = maptree(swap(t), f)$
$tipcount(t) = labelcount(t) + 1$
$\lim(\lambda x.f_1(x), a, l_1) \wedge \lim(\lambda x.f_2(x), a, l_2) \rightarrow$
$\quad \lim(\lambda x.f_1(x) + f_2(x), a, l_1 + l_2)$
$\lim(\lambda x.f_1(x), a, l_1) \wedge \lim(\lambda x.f_2(x), a, l_2) \rightarrow$
$\quad \lim(\lambda x.f_1(x) - f_2(x), a, l_1 - l_2)$

Theorems marked (*) are from [3]. All but the LIM theorems are without lemmas.

Table 4. Some theorems which can be proved by the system

5 RELATED AND FUTURE WORK

Hutter has also developed a calculus for manipulating annotated terms which is implemented in the INKA system [5]. His annotated terms, called *C-terms*, also have a notion of "colour" similar to that used here. Hutter has given algorithms for unifying together annotated terms and for substitution into annotated terms; these are used to perform rewriting of annotated and coloured terms. Although this calculus can deal with multiple colours, INKA uses only two (denoting the wave-hole and wave-front) since the problem of mixing of skeletons has not been identified. Therefore, Hutter did not use his colours for coloured rippling. We believe, however, that it might be possible to map coloured rippling into the C-term calculus.

There are many possible directions for further work. We only have space here to list a few of them.

Difference matching. Annotations in the consequent of the LIM$^+$ and LIM$^-$ theorem were added by hand. Ideally, wave annotation in the consequent should be added automatically by the system, as in inductive proofs. The *difference matching* of [1] adds monochromatic wave annotations to terms. We plan to use this procedure to add coloured wave annotation. The idea is to combine together *compatible* monochromatic answers from difference matching to give coloured annotations. This algorithm is quite similar to the one described in §3 and presented in [8], however, it has not been implemented yet.

Loss of colours. Losing colours dynamically during rippling is expected to enable the system to prove more difficult theorems [8]. However, this has a potential exponential cost since we would need to consider every possible weakening of colours. If a wave-hole has n colours, there are $O(2^n)$ possible weakenings. This may considerably worsen the efficiency of the system (though n is usually small). In addition, we intend to investigate when colour weakening should be applied.

$$(\boxed{\underline{U_1}_X + \underline{U_2}_Y}^{\uparrow}) - (\boxed{\underline{V_1}_X + \underline{V_2}_Y}^{\uparrow}) \Rightarrow \boxed{(\underline{U_1 - V_1}_X) + (\underline{U_2 - V_2}_Y)}^{\uparrow} \tag{11}$$

$$|\boxed{\underline{U}_X + \underline{V}_Y}^{\uparrow}| < E \Rightarrow \boxed{\underline{|U|}_X + \underline{|V|}_Y}^{\uparrow} < E$$

$$\boxed{\underline{U}_X + \underline{V}_Y}^{\uparrow} < W \Rightarrow \boxed{\underline{U < \boxed{\frac{\underline{\underline{W}}_X}{2}}^{\downarrow}}_X \wedge \underline{V < \boxed{\frac{\underline{\underline{W}}_Y}{2}}^{\downarrow}}_Y}^{\uparrow}$$

$$Q \rightarrow \boxed{\underline{P_1}_X \wedge \underline{P_2}_Y}^{\uparrow} \Rightarrow \boxed{\underline{Q \rightarrow P_1}_X \wedge \underline{Q \rightarrow P_2}_Y}^{\uparrow} \tag{12}$$

$$\forall x.\boxed{\underline{P_1}_X \wedge \underline{P_2}_Y}^{\uparrow} \Rightarrow \boxed{\underline{\forall x.P_1}_X \wedge \underline{\forall x.P_2}_Y}^{\uparrow}$$

$$\exists \delta.\Psi(U < \delta \rightarrow \boxed{\underline{P_1}_X \wedge \underline{P_2}_Y}^{\uparrow}) \Rightarrow \boxed{\underline{\exists \delta.\Psi(U < \delta \rightarrow P_1)}_X \wedge \underline{\exists \delta.\Psi(U < \delta \rightarrow P_2)}_Y}^{\uparrow} \tag{13}$$

$$0 < \epsilon \rightarrow \boxed{\underline{P}_X \wedge \underline{Q}_Y}^{\uparrow} \Rightarrow \boxed{\underline{0 < \tfrac{\epsilon}{2} \rightarrow P}_X \wedge \underline{0 < \tfrac{\epsilon}{2} \rightarrow Q}_Y}^{\uparrow}$$

Table 3. Wave-rules for LIM$^+$

N-ary trees. This paper has described coloured rippling using binary trees. Binary trees can be generalized to n-ary trees. Since binary trees have a fixed number of branches, it is relatively easy to define the tree induction method. N-ary trees are more problematic since it is not possible to determine in advance the number of branches.

Decomposition of the antecedent into hypotheses. In §3, CNF is used to convert the antecedent into hypotheses in the non-inductive method. However, this may give many too small hypotheses. Techniques for decomposing the antecedent into hypotheses is left for further work (one possibility is to use difference matching, with the aim of removing the difference between the antecedent and the consequent).

6 CONCLUSIONS

This paper has demonstrated the successful application of rippling to data structures with multiple recursive arguments. The presence of multiple recursive arguments may upset monochromatic rippling because these arguments can become mixed. The key idea to solve this problem is to add colour annotation to wave-holes to distinguish between the different arguments. This idea can also be applied to challenging non-inductive theorems like limit theorems. Our experiments have given us considerable confidence in the applicability of this extension to rippling; the theorems about binary trees and limits were proved in a uniform way which should also be applicable to many other examples.

Acknowledgements T. Yoshida was supported by the Rotary Scholarship Program. Grant number SERC GR/H/23610 supported I. Green and computing equipment. T. Walsh was supported by a SERC postdoctoral fellowship and a Human Capital and Mobility fellowship. D. Basin was funded by the German Ministry for Research and Technology (BMFT) under grant ITS 9102. The responsibility for the contents of this publication lies with the authors.

REFERENCES

[1] D. Basin and T. Walsh, 'Difference unification', in *Proceedings of the 13th IJCAI*. International Joint Conference on Artificial Intelligence, (1993). Also available as Technical Report MPI-I-92-247, Max-Planck-Institute für Informatik.

[2] W.W. Bledsoe, 'Challenge problems in elementary calculus', *Journal of Automated Reasoning*, **6**(3), 341–359, (1990).

[3] R.S. Boyer and J.S. Moore, *A Computational Logic*, Academic Press, 1979. ACM monograph series.

[4] A. Bundy, A. Stevens, F. van Harmelen, A. Ireland, and A. Smaill, 'Rippling: A heuristic for guiding inductive proofs', *Artificial Intelligence*, **62**, 185–253, (1993). Also available from Edinburgh as DAI Research Paper No. 567.

[5] D. Hutter, 'Guiding inductive proofs', in *10th International Conference on Automated Deduction*, ed., M.E. Stickel, pp. 147–161. Springer-Verlag, (1990). Lecture Notes in Artificial Intelligence No. 449.

[6] A. Stevens, 'A rational reconstruction of Boyer and Moore's technique for constructing induction formulas', in *The Proceedings of ECAI-88*, ed., Y. Kodratoff, pp. 565–570. European Conference on Artificial Intelligence, (1988). Also available from Edinburgh as DAI Research Paper No. 360.

[7] T. Walsh, A. Nunes, and A. Bundy, 'The use of proof plans to sum series', in *11th Conference on Automated Deduction*, ed., D. Kapur, pp. 325–339. Springer Verlag, (1992). Lecture Notes in Computer Science No. 607. Also available from Edinburgh as DAI Research Paper 563.

[8] Tetsuya Yoshida, *Coloured Rippling*, Master's thesis, Dept. of Artificial Intelligence, Edinburgh, 1993.