

Planning and Patching Proofs: Exercise Solutions

Lucas Dixon and Alan Bundy

August 26, 2010

1 Introduction

This is the solution sheet for the problems in *Planning and Patching Proofs: Exercises*.

2 Your First Ripple: mapping a function over a list

The definition of *append*:

$$\text{nil} @ l = l \quad (1)$$

$$(h :: t) @ l = h :: (t @ l) \quad (2)$$

The definition of *map*:

$$\text{map}(f, \text{nil}) = \text{nil} \quad (3)$$

$$\text{map}(f, a :: l) = f(a) :: \text{map}(f, l) \quad (4)$$

Theorem: $\text{map}(f, l) @ \text{map}(f, k) = \text{map}(f, l @ k)$

Proof.

1. Apply the induction scheme to l to get the following base- and step-case subgoals:

Base-Case: $\text{map}(f, \text{nil}) @ \text{map}(f, k) = \text{map}(f, \text{nil} @ k)$

Step-Case: For a fixed h and t assume the induction hypothesis:

$$\forall k. \text{map}(f, t) @ \text{map}(f, k) = \text{map}(f, t @ k)$$

and prove the step-case goal for a fixed k :

$$\text{map}(f, h :: t) @ \text{map}(f, k) = \text{map}(f, h :: t @ k)$$

2. The proof of the base-case follows by simplification, using (3) once and (1) twice.
3. The rippling goal of the step-case is:

$$\begin{aligned} \text{map}(f, h :: t) @ \text{map}(f, k) &= \text{map}(f, h :: t @ k) \\ &\Downarrow \text{by (4)} \\ (f(h) :: \text{map}(f, t)) @ \text{map}(f, k) &= \text{map}(f, h :: t @ k) \\ &\Downarrow \text{by (2)} \\ f(h) :: (\text{map}(f, t) @ \text{map}(f, k)) &= \text{map}(f, h :: t @ k) \\ &\Downarrow \text{by (2)} \\ f(h) :: (\text{map}(f, t) @ \text{map}(f, k)) &= \text{map}(f, h :: t @ k) \\ &\Downarrow \text{by (4)} \\ f(h) :: (\text{map}(f, t) @ \text{map}(f, k)) &= f(h) :: \text{map}(f, t @ k) \end{aligned}$$

Rippling is *blocked*. Weak fertilisation gives the subgoal:

$$f(h) :: \text{map}(f, t @ k) = f(h) :: \text{map}(f, t @ k)$$

and this an instance of reflexivity of equality. Note that there are several ways the proof may be completed. Weak fertilisation would be applied to the right hand side of the subgoal instead of the left. Also, given the rule for argument congruence:

$$P(X) = P(Y) \iff X = Y$$

then rippling can perform a final ripple-step that removes all wave fronts, and thus concludes the proof by observing that the the final subgoal is identical to the induction hypothesis. □

3 A Critic: lemma calculation

The defining equations for additon and multiplication are:

$$0 + n = n \tag{5}$$

$$Suc(m) + n = Suc(m + n) \tag{6}$$

$$0 * n = 0 \tag{7}$$

$$Suc(m) * n = n + Suc(m * n) \tag{8}$$

Theorem: $(x * y) * z = x * (y * z)$

Proof.

1. Induction on x gives the following base- and step-case subgoals:

Base-Case: $(0 * y) * z = 0 * (y * z)$

Step-Case: For a fixed x , assume the induction hypothesis: $\forall y, z. (x * y) * z = x * (y * z)$ and prove the step-case goal for fixed y and z :

$$(Suc(x) * y) * z = Suc(x) * (y * z)$$

2. The proof of the base-case follows by simplification, using (5) twice.
3. The rippling goal of the step-case is:

$$\begin{aligned} (Suc(x) * y) * z &= Suc(x) * (y * z) \\ &\Downarrow \text{by (8)} \\ (y + (x * y)) * z &= Suc(x) * (y * z) \\ &\Downarrow \text{by (8)} \\ (y + (x * y)) * z &= (y * z) + (x * (y * z)) \end{aligned}$$

Rippling is *blocked*. Weak fertilisation gives the subgoal:

$$(y + (x * y)) * z = (y * z) + ((x * y) * z)$$

4. Lemma calculation, with common subterm generalisation, conjectures the lemma:

$$(u + v) * w = (u * w) + (v * w)$$

The proof of this lemma by induction and rippling is given below.

5. The calculated lemma is then used to prove the final subgoal of the step-case, concluding the proof. □

Lemma: $(u + v) * w = (u * w) + (v * w)$

Proof.

1. Induction on u gives the following base- and step-case subgoals:

Base-Case: $(0 + v) * w = (0 * w) + (v * w)$

Step-Case: For a fixed u , assume the induction hypothesis:

$$\forall v, w. (u + v) * w = (u * w) + (v * w)$$

and prove the step-case goal for fixed v and w :

$$(Suc(u) + v) * w = (Suc(u) * w) + (v * w)$$

2. The proof of the base-case follows by simplification, using (7) and (5) twice.
3. The rippling goal of the step-case is:

$$\begin{aligned} (Suc(u) + v) * w &= (Suc(u) * w) + (v * w) \\ &\Downarrow \text{by (6)} \\ (Suc(u + v)) * w &= (Suc(u) * w) + (v * w) \\ &\Downarrow \text{by (8)} \\ w + ((u + v) * w) &= (Suc(u) * w) + (v * w) \\ &\Downarrow \text{by (8)} \\ w + ((u + v) * w) &= (w + (u * w)) + (v * w) \end{aligned}$$

Rippling is *blocked*. Weak fertilisation gives the subgoal:

$$w + ((u * w) + (v * w)) = (w + (u * w)) + (v * w)$$

4. Lemma calculation, with common subterm generalisation, conjectures the lemma:

$$x + (y + z) = (x + y) + z$$

The proof of this lemma by induction and rippling is given below.

5. The calculated lemma is then used to prove the final subgoal of the step-case, concluding the proof. □

Lemma: $x + (y + z) = (x + y) + z$

Proof.

1. Induction on x gives the following base- and step-case subgoals:

Base-Case: $0 + (y + z) = (0 + y) + z$

Step-Case: For a fixed x , assume the induction hypothesis:

$$\forall y, z. x + (y + z) = (x + y) + z$$

and prove the step-case goal for fixed y and z :

$$Suc(x) + (y + z) = (Suc(x) + y) + z$$

2. The proof of the base-case follows by simplification, using (5) twice.
3. The rippling goal of the step-case is:

$$\begin{aligned}
\mathit{Suc}(x) + (y + z) &= (\mathit{Suc}(x) + y) + z \\
&\Downarrow \text{by (6)} \\
\mathit{Suc}(x + (y + z)) &= (\mathit{Suc}(x) + y) + z \\
&\Downarrow \text{by (6)} \\
\mathit{Suc}(x + (y + z)) &= \mathit{Suc}(x + y) + z \\
&\Downarrow \text{by (6)} \\
\mathit{Suc}(x + (y + z)) &= \mathit{Suc}((x + y) + z)
\end{aligned}$$

Rippling is *blocked*. Weak fertilisation gives the subgoal:

$$\mathit{Suc}((x + y) + z) = \mathit{Suc}((x + y) + z)$$

and this an instance of reflexivity of equality. Like the proof that map distributes over append, there are several ways this proof can be completed.

□