Towards a ‘Repertoire of Reasons’

Alan Bundy

April 25, 2011

Examples of ‘Why’ in Theorem Proving

- The productive use of failure.
- Examples of failed proof attempts and how to patch them.
- Use simplest example that illustrates point.
- Drawn from our induction and equation solving experience,
  - but rippling annotation avoided.

Why Introduce a Lemma?

Conjecture: $\forall l{:}list(\tau).\ rev(rev(l)) = l$

Available rewrite rule: $rev(h :: t) \Rightarrow rev(t) @ (h :: nil)$

Stuck step case:

$$\begin{aligned} rev(rev(t)) = t \ \vdash \ & rev(rev(h :: t)) = h :: t \\ \vdash \ & rev(rev(t) @ (h :: nil)) = h :: t \end{aligned}$$

Missing lemma scheme: $rev(l @ k) = F(rev(l), k, l)$

Missing lemma: $rev(l @ k) = rev(k) @ rev(l)$

Why Generalise Apart?

Conjecture: $\forall n{:}\mathbb{N}.\ n + (n + n) = (n + n) + n$

Available rewrite rule: $s(x) + y \Rightarrow s(x + y)$

Stuck step case:

$$\begin{array}{ll} \ldots & \vdash \quad s(n) + (s(n) + s(n)) = (s(n) + s(n)) + s(n) \\ & \vdash \quad s(n) + (s(n + s(n))) = s(n + s(n)) + s(n) \\ & \vdash \quad s(n + s(n + s(n))) = s((n + s(n)) + s(n)) \end{array}$$

Generalised conjecture: $\forall m, n{:}\mathbb{N}.\ m + (n + n) = (m + n) + n$

Successful step case:

$$\begin{array}{ll} \ldots & \vdash \quad s(m) + (n + n) = (s(m) + n) + n \\ & \vdash \quad s(m + (n + n)) = s(m + n) + n \\ & \vdash \quad s(m + (n + n)) = s((m + n) + n) \\ & \vdash \quad s((m + n) + n) = s((m + n) + n) \end{array}$$

Why Generalise Subterms?

Conjecture: $\forall l{:}list(\tau).\ rev(rev(l)) = l$

Apply induction hypothesis:

$$
\begin{array}{rll}
rev(rev(t)) = t & \vdash & rev(rev(h :: t)) = h :: t \\
 & \vdash & rev(rev(t)@(h :: nil)) = h :: t \\
 & \vdash & rev(rev(t)@(h :: nil)) = h :: rev(rev(t))
\end{array}
$$

New Conjecture:

$$\forall h{:}\tau, l{:}list(\tau).\ rev(rev(t)@(h :: nil)) \quad = \quad h :: rev(rev(t))$$

Generalise subterm: replace $rev(t)$ with k.

$$\forall h{:}\tau, k{:}list(\tau).\ rev(k@(h :: nil)) \quad = \quad h :: rev(k)$$

Why Generalise Accumulators?

Conjecture: $\forall l{:}list(\tau).\ rev(l) = qrev(l, nil)$

Available rewrite rules:

$$\begin{aligned} rev(h :: t) &\Rightarrow rev(t) @ (h :: nil) \\ qrev(h :: t, l) &\Rightarrow qrev(t, h :: l) \end{aligned}$$

Stuck step case:

$$\begin{aligned} \ldots &\vdash rev(h :: t) = qrev(h :: t, nil) \\ \ldots &\vdash rev(t) @ (h :: nil) = qrev(t, h :: nil) \end{aligned}$$

Generalised conjecture: $\forall k, l{:}list(\tau).\ rev(l) @ k = qrev(l, k)$

Successful step case:

$$\begin{aligned} & rev(t) @ K = qrev(t, K) \\ & \quad \vdash rev(h :: t) @ k = qrev(h :: t, k) \\ & \quad \vdash rev(t) @ (h :: nil) @ k = qrev(t, h :: k) \\ & \quad \vdash rev(t) @ (h :: k) = qrev(t, h :: k) \end{aligned}$$

Why Change Induction Rules?

Conjecture: $\forall n{:}\mathbb{N}.\ even_m(n) \vee even_m(s(n))$

Available rewrite rules:

$$\begin{aligned} even_m(s(n)) &\Rightarrow odd_m(n) \\ odd_m(s(n)) &\Rightarrow even_m(n) \end{aligned}$$

Stuck step case:

$$\begin{aligned} \ldots &\vdash even_m(s(n)) \vee even_m(s(s(n))) \\ &\vdash odd_m(n) \vee odd_m(s(n)) \\ &\vdash odd_m(n) \vee even_m(n) \end{aligned}$$

Revised step case:

$$\begin{aligned} \ldots &\vdash even_m(s(s(n))) \vee even_m(s(s(s(n)))) \\ &\vdash odd_m(s(n)) \vee odd_m(s(s(n))) \\ &\vdash even_m(n) \vee even_m(s(n)) \end{aligned}$$

Why Conjoin Mutual Duals?

Conjecture: $\forall n{:}\mathbb{N}.\ even_m(n) \vee even_m(s(n))$

Available rewrite rules:

$$\begin{aligned} even_m(s(n)) &\Rightarrow odd_m(n) \\ odd_m(s(n)) &\Rightarrow even_m(n) \end{aligned}$$

Generalised conjecture:

$$\forall n{:}\mathbb{N}.\ [even_m(n) \vee even_m(s(n))] \wedge [odd_m(n) \vee odd_m(s(n))]$$

Successful step case:

$$\begin{aligned} &[even_m(n) \vee even_m(s(n))] \wedge [odd_m(n) \vee odd_m(s(n))] \\ &\quad \vdash \quad [even_m(s(n)) \vee even_m(s(s(n)))] \wedge [odd_m(s(n)) \vee odd_m(s(s(n)))] \\ &\quad \vdash \quad [odd_m(n) \vee odd_m(s(n))] \wedge [even_m(n) \vee even_m(s(n))] \end{aligned}$$

Why Generalise to Decidable Form?

Conjecture:

$$\forall k, l{:}\mathbb{R}, a{:}array(\mathbb{R}).\ l \leq min(a) \wedge 0 < k \implies l < max(a) + k$$

False generalised conjecture (linear arithmetic):

$$\forall k, l, min, max{:}\mathbb{R}.\ l \leq min \wedge 0 < k \implies l < max + k$$

Counter-example: $l = min = 2,\ k = 1,\ max = 0.$

True conditional conjecture:

$$\forall k, l, min, max{:}\mathbb{R}.\ l \leq min \wedge 0 < k \wedge min \leq max \implies l < max + k$$

Why Piecewise Fertilise Step Cases?

Conjecture:

$$Single\_Occ(x, l = r) \wedge Posn(x, l, p) \wedge Isolate(p, l = r, x = a) \implies Solve(l = r, x, x = a)$$

Induction conclusion:

$$Single\_Occ(X, L = R) \wedge Posn(X, L, H :: P) \wedge Isolate(H :: P, L = R, X = A) \implies Solve(L = R, X, X = A)$$

Piecewise fertilization:

Use	**In the proof of**
$Single\_Occ(X, L = R)$	$Single\_Occ(x, l = r)$
$Posn(X, L, H :: P)$	$Posn(x, l, P)$
$Isolate(H :: P, L = R, X = A)$	$Isolate(P, l = r, x = a)$
$Solve(l = r, x, x = a)$	$Solve(L = R, X, X = A)$

Why Shake but don't Stir?

Conjecture: $\forall t : Tree(\tau).\ Maxht(t) \geq Minht(t)$
where $Tree(\tau) ::= Leaf(\tau) \mid Node(Tree(\tau), Tree(\tau))$.

Step case:

$$Maxht(l) \geq Minht(l), \qquad Maxht(r) \geq Minht(r)$$
$$\vdash$$
$$Max(Maxht(l), Maxht(r)) \quad \geq \quad Min(Minht(l), Minht(r))$$

Rival rewrite rules:

$$Max(u_1, u_2) \geq Min(v_1, v_2) \quad \Rightarrow \quad u_1 \geq v_1 \wedge u_2 \geq v_2$$
$$Max(u_1, u_2) \geq Min(v_1, v_2) \quad \Rightarrow \quad u_1 \geq v_2 \wedge u_2 \geq v_1$$

Why Isolate Unknowns?

Equation with one unknown occurrence:

$$e^{sin(x)} - 1 = 0$$

Isolate x:

$$\begin{aligned} e^{sin(x)} &= 0 + 1 \\ sin(x) &= log_e(1) \\ x &= arcsin(0) \end{aligned}$$

Why Collect Unknowns?

Equation with two unknown occurrences:

$$sin(x).cos(x) - 1 \quad = \quad 0$$

Collecting xs:

$$\frac{sin(2.x)}{2} - 1 \quad = \quad 0$$

Why Attract Unknowns?

Equation with distant unknown occurrences:

$$cos^2(x) + 3 = sin^2(x) + 4$$

Attracting then collecting xs:

$$\begin{aligned} cos^2(x) + 3 - sin^2(x) &= 4 \\ cos^2(x) - sin^2(x) &= 4 - 3 \\ cos(2.x) &= 1 \end{aligned}$$

Why Homogenise Equations

Equation from diverse areas:

$$cos^2(x) + 2 = 2.sin(x) + 4$$

Equation homogenised then areas separated:

$$1 - sin^2(x) + 2 = 2.sin(x) + 4$$

$$\begin{aligned} 1 - y^2 + 2 &= 2.y + 4 \\ sin(x) &= y \end{aligned}$$

Compare homogenisation to generalisation to decidable form.

Conclusion

- Analysis of proof failure can suggest patch.
- Successful analysis requires expectation of proof direction.
- This localises assignment of blame.
- Examples shown arose from initial human analysis.
- Could this be automated?