Inductive proofs

Friday, December 7, 2018 11:48 AM

1) inductive proofs for predicates even n & odd n

1.a $\dfrac{}{\text{even } z}$ 2.a $\dfrac{}{\text{odd } s(z)}$

1.b $\dfrac{\text{even } n}{\text{even } s(s(n))}$ 2.b $\dfrac{\text{odd } n}{\text{odd } s(s(n))}$

(2) inductive proofs $\forall n$ even $n \to$ odd $s(n)$ and $\forall n$ odd $n \to$ even $s(n)$

(a) $\forall n$ even $\to$ odd $s(n)$
induction on even n

case (a.1) $n = z$
even z , by rule 2.a odd s(z)

case (a.2) $n = s(s(n'))$, even n'

by IH on even $n' \to$ odd $s(\underbrace{s(s(n'))}_{=n})$

odd s(n)

$\forall n$ odd $n \to$ even $s(n)$
by induction on odd n

case (2.1) $n = s(z)$
odd s(z)
by IH , even s(s(z))
by rule 1.b even $s(\underline{s(z)})$: even z

case (2.2) $n = s(s(n'))$, odd n'.

by IH, even $s(\underbrace{s(s(n'))}_{=n})$, even s(n)

2.1 $\forall n$ $\underline{\text{even } n \to \text{odd } s(n)}$

case (1) $n = z$
even z = odd s(z)
by b.1

case(2) $n = s(n')$ (goal: odd s(s(n')))

$\underline{\text{even } s(n')}$ = $\underline{\text{odd } s(s(n'))}$ by IH

2.2 $\forall n$ odd $n \to$ even s(n)
case (a) $n = s(z)$
$\underline{\text{odd}}\ s(z)$ = even $\underline{s(s(z))}$ by IH

case (b) $n = s(s(n'))$ (goal: odd n' = even s(n'))

odd s(s(n')) = even s(s(s(n'))) by IH
= $\underline{\text{even } s(n')}$

(3) $\forall n$, even n $\vee$ odd n

$\underline{\text{case}}$(a) $n = z$
even z $\vee$ odd z

case (b) $n = s(n')$

$\underline{\text{even } s(n')}$ $\vee$ odd s(n')
case (b.1)
even s(n') = odd n' by rule 2.1

case b.2 odd s(n') = even n' by rule 2.2

(3)

$\forall n$ even n $\vee$ odd n

induction on n then $n = z$ or $n = s(n')$
case (1) $n = z$ [Goal :- even z $\vee$ odd z]

by apply IH, even z $\vee$ odd z
by rule (1.a) even z

case (2) $n = s(n')$ [Goal: even s(n') $\vee$ odd s(n')]
, n' nat

by apply IH even n' $\vee$ odd n'

case (2.1) even n' then odd s(n') by rule
$\forall n$ even $n \to$ odd s(n')

case (2.2) odd n' then even s(n') by rule
$\forall n$, n nat, odd $n \to$ even s(n)

plus n m x $\underline{n+m} = x$

(4) plus 1 : plus z m m
plus 2 : plus s(n') m S(x) :- plus n' m x

plus s(n) m = s(plus n m)

Times 1 : times z n z
Times 2 : times s(n) m x : plus (Times n' m) m
: Times n' m x_1
plus x_1 m x

(n+1) $\cdot$ m
nm + m

(4) definition of the operations n+m & n.m by recursion on n

(P1) $\dfrac{}{\text{plus } z\ m\ m}$

(P2) $\dfrac{\text{plus } n\ m\ x}{\text{plus } S(n)\ m\ S(x)}$

(P3) $\dfrac{}{\text{plus } n\ z\ n}$

(P4) $\dfrac{\text{plus } n\ m\ x}{\text{plus } n\ s(m)\ S(x)}$

(T1) times z m z
$\boxed{s(n) \cdot m = n \cdot m + m}$
(T2) times s(n) m y :- times n m y_1 , plus y_1 m y

(T3) times ~~m~~ z z

(T4) times n s(m) y :- times n m y_1 , plus y_1 n y

(5) $\forall n\ \forall m\ \forall l$ $n + (m+l) = (n+m)+l$

induction n
case (a) $n = z$ goal : z + (m+l)
z + (m+l)
by rule P1 m+l

case (b) $n = s(n')$
by IH $n' + (m+l) = (n'+m)+l$

$S(n') + (m+l)$
by rule P2 $S(n'+(m+l))$

distributive

(5) $\forall n\ \forall m\ \forall l$ $n + (m+l) = (n+m)+l$

case (a) $n = z$ ((z+m)+l)
$\underline{z + (m+l)}$ = m+l by rule plus1 z+m = m
= z+m+l by rule plus 1
= $\underline{(z+m)}$+l by IH

case (b) $n = s(n')$ (s(n')+(m+l) = (s(n')+m)+l)

s(n')+(m+l) = s(n'+(m+l)) plus 3
= s($\underline{(n'+m)+l}$) by IH
= $\underline{(s(n')+m)+l}$ plus 3

$\boxed{\text{list} ::= \underline{\text{nil}} \mid \underline{\text{cons}(n;l)}}$

lenght 1 : |nil| = z
lenght 2 : |cons(n;l)| = s(|l|)

by applying IH $s(n'+m)+l$

by rule p2 $= s(n'+m)+l$
rule p2 $= (s(n')+m)+l$

⑥ $\forall n \; n+Z = n$ and
$\forall n \forall m \; n+s(m) = s(n+m)$

(a) $\forall n \quad n+Z = n$
induction on n then
case (a.1) $n = Z$
$Z+Z = Z$ by rule (P1)

case (a.2) $n = s(n')$
by IH, $n'+Z = n'$

$s(n')+Z = s(n'+Z)$ rule (P2)
$= s(n')$ by IH

(b) $\forall n \forall m \; n+s(m) = s(n+m)$

induction on 'n'

case (b.1) $n = Z$
$Z + s(m) = s(m)$ by rule (P1)

case (b.2) $n = s(n')$ $\boxed{\text{goal: } s(s(n')+m)}$
$s(n')+s(m)$
by P2 $s(n'+s(m))$
by IH $s(s(n'+m))$
by rule P2 $s(s(n')+m)$

⑦ $\forall n \forall m \quad n+m = m+n$

induction on n

Case (a) $n = Z$
$Z+m = m$ by rule (P1)
$= m+Z$ by 6(a) $(m+Z = m)$

Case (b) $n = s(n')$ $\boxed{\text{goal:- } s(n')+m = m+s(n')}$

$s(n')+m = s(n'+m)$ by rule 4(a) :- $s(n)+m = s(n+m)$
$= s(m+n')$ commutativity
$= m+s(n')$ by rule 6(b) :- $s(m+n) = m+s(n)$.

⑨ length $|l|$, concatenation $l_1 \cdot l_2$
reverse $rev(l)$

(l1) $len(nil) = Z$

(l2) $len(cons(n,l)) = s(len(l))$

(C1) $nil \cdot l = l$
(C2) $cons(n,l_1) \cdot l_2 = cons(n, l_1 \cdot l_2)$

(R1) $rev(nil) = nil$
(R2) $rev(cons(n;l)) = rev(l) \cdot cons(n;nil)$

⑩ prove $|l_1 \cdot l_2| = |l_1| + |l_2|$
and $|rev(l)| = |l|$

ⓐ $|l_1 \cdot l_2| = |l_1| + |l_2|$

induction l_1

Case (a) : $l_1 = nil$

$|nil \cdot l_2| = |l_2|$ by (C1)

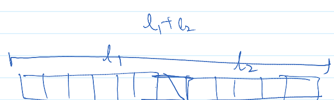
$= Z+|l_2|$ by rule (P1)

$= |nil| + |l_2|$ by rule (l1)

Case (b) $l_1 = cons(n;l_1')$ goal: $|cons(n,l')| + |l_2|$

$|cons(n;l') \cdot l_2| = |cons(n; l' \cdot l_2)|$ by rule (C2)

$= s(|l' \cdot l_2|)$ by Rule (l2)



⑨ $\forall l_1 \forall l_2 \quad |l_1 \cdot l_2| = |l_1| + |l_2|$

Case (a) $l_1 = nil$ (goal:- $Z + |l_2|$)

$|nil \cdot l_2| = |l_2|$ by rule C1

$= Z+|l_2|$ by rule plus1 $(Z+m = m)$

$= |nil| + |l_2|$ by rule len1

☒

Case (b) $l_1 = cons(n;l_1')$ (goal: $|cons(n;l_1) \cdot l_2| = |cons(n;l_1)| + |l_2|$)

$|cons(n;l_1) \cdot l_2| = |cons(n; l_1 \cdot l_2)|$ by rule C2

$= s(|l_1 \cdot l_2|)$ by l2

$= s(|l_1| + |l_2|)$ by IH

$= s(|l_1|) + |l_2|$ by rule P3 $\boxed{s(m+n) = s(m)+n}$

$= |cons(n;l_1)| + |l_2|$ by rule (l2)

☒

⑪(a) $|rev(l)| = |l|$

Case (a) $l = nil$ $(|rev(nil)| = |nil|)$

$|rev(nil)| = |nil|$ by rule r1

Case (b) $l = cons(n;l)$ (goal: $|rev(cons(n;l))| = |cons(n;l)|$)

$\boxed{|rev(cons(n;l))|} = |rev(l) \cdot cons(n;nil)|$ by rule rev2

$|l_1 \cdot l_2| = |l_1| + |l_2|$

$= |rev(l)| + |cons(n;nil)|$

$= s(|l'\cdot l_2|)$ by Rule (l2)

$= s(|l'| + |l_2|)$ by IH

$= s(|l'|) + |l_2|$ by (P2)

$= |cons(n; l')| + |l_2|$ by (l2)

(b) $|rev(l)| = |l|$

induction l

Case (b.1) $l = nil$

$|rev(nil)| = |nil|$ by rule (R1) $= 0$ by rule (l1)

Case (b.2) $l = cons(n; l')$ goal: $|rev(cons(n; l'))| = s(|l'|)$

$|rev(cons(n, l'))| = |rev(l') \cdot cons(n; nil)|$ by (R2)

$= |rev(l')| + |cons(n; nil)|$ by (l2)

$= |rev(l')| + s(z)$ by (l2)

$s(|rev(l')|)$ by (P2)

$= s(|l'|)$ by IH

$= |rev(l)| + |cons(n; nil)|$

$= |l| + |cons(n, nil)|$ by IH

$= |cons(n; nil)| + |l|$ by associ $a+b = b+a$

$= |cons(n, nil) \cdot l|$ by $|l_1 \cdot l_2| = |l_1| + |l_2|$

$= |cons(n; nil \cdot l)|$ by c2

$= |cons(n, l)|$ by C1

11) $rev(l_1 \cdot l_2) = rev(l_2) \cdot rev(l_1)$

and $rev(rev(l)) = l$

a) $rev(l_1 \cdot l_2) = rev(l_2) \cdot rev(l_1)$

by induction l_1

Case (a.1) :: $l_1 = nil$ $\boxed{\text{goal } rev(nil \cdot l_2) = rev(l_2) \cdot rev(nil)}$

$rev(nil \cdot l_2) = rev(l_2)$ by rule (c1)

$= rev(l_2) \cdot nil$ by lemma $l_1 \cdot nil = l_1$

$= rev(l_2) \cdot rev(nil)$ by rule (R1)

Case (a.2) $l_1 = cons(n; l_1')$

goal: $rev(cons(n; l_1') \cdot l_2) = rev(l_2) \cdot rev(cons(n; l_1'))$

$rev(cons(n; l_1') \cdot l_2) = rev(cons(n; l_1' \cdot l_2))$ by rule (c2)

$= rev(l_1' \cdot l_2) \cdot (cons(n; nil))$ by rule (R2)

$= rev(l_1') \cdot rev(l_2) \cdot (cons(n; nil))$ by (c2)

$= rev(l_2) \cdot rev(l_1') \cdot (cons(n; nil))$ by apply IH

$= rev(l_2) \cdot rev(cons(n; l_1'))$ by (R2)

(b) $rev(rev(l)) = l$

by induction on l

case (b.1) . $l = nil$ $\boxed{\text{goal} = rev(rev(nil)) = nil}$

$rev(rev(nil)) = rev(nil)$ by (R1)

$= nil$ by rule (R1)

Case (b.2) $l = cons(n; l')$ $\boxed{\text{goal } rev(rev(cons(n; l'))) = cons(n; l')}$

$rev(rev(cons(n; l'))) =$

$= rev(rev(l') \cdot cons(n; nil))$ by (R2)

$= rev(cons(n; nil)) \cdot rev(rev(l'))$ by (R2)

$= rev(nil) \cdot cons(n; nil) \cdot rev(rev(l'))$ by (R2)

$= nil \cdot cons(n; nil) \cdot rev(rev(l'))$ by (R1)

$= cons(n; nil \cdot) \cdot rev(rev(l'))$ by (C1)

= nil · cons(n; nil) · rev(rev(l')) by (R1)

= cons(n; nil·) · rev(rev(l')) by (C1)

= cons(n; nil) · (l') by IH

= cons(n; l') by (C1)

(12) tree t ::= empty | node(n; t1; t2)

(t1) size(empty) = z

(t2) size(node(n; t1; t2)) = S(size(t1) + size(t2))

(t3) height(empty) = z

(t4) height(node(n; t1; t2)) = S(max(height(t1), height(t2)))

(13) size(rev(tree)) = size(tree)

rev1 rev(empty) = empty

rev2 rev(node(n; t1; t2)) = node(n; rev(t2); rev(t1))

(13) ∀t tree t → size(rev(t)) = size(t)

apply induction on t

case (a) t = empty $\boxed{\text{goal :- size(rev(empty)) = size(empty)}}$

size(rev(empty)) = size(empty) by rule (rev1)

case (b) : t = node(n; t1; t2) $\boxed{\text{goal :- size(rev(node(n; t1; t2))) = size(node(n; t1; t2))}}$

size(rev(node(n; t1; t2))) = size(node(n; rev(t2); rev(t1))) by rule (rev2)

= S(size(rev(t2)) + size(rev(t1))) by rule (t2)

= S(size(t2) + size(t1)) by apply IH

= S(size(t1) + size(t2)) by commutativity

= size(node(n; t1; t2)) by (t2)

(b) height(rev(t)) = height(t)

induction t

case (b.1) t = empty

height(rev(empty)) = height(empty) by rule (rev1)

case (b.2) t = node(n; t1; t2) [goal: height(rev(node(n; t1; t2))) = height(node(n; t1; t2))]

height(rev(node(n; t1; t2)))

= height(node(n; rev(t2); rev(t1))) by (rev2)

= S(max(height(rev(t2)); height(rev(t1)))) by (t3)

= S(max(height(t2); height(t1))) by IH

= S(max(height(t1); height(t2))) by lemma max(n1; n2) = max(n2; n1)

= height(node(n; t1; t2)) by rule (t3)

(13) preorder = <root> <left> <right>

Inorder = <left> <root> <right>

postorder = <left> <right> <root>

(P1) pre(empty) = nil

(P2) pre(node(n; t1; t2)) = cons(n; nil) · (pre(t1) · pre(t2))

= cons(n; pre(t1) · pre(t2))

(In1) In(empty) = nil

(In2) In(node(n; t1; t2)) = In(t1) · (cons(n; nil) · In(t2))

= In(t1) · cons(n; In(t2))

(PO1) post(empty) = nil

(PO2) post(node(n; t1; t2)) = post(t1) · (post(t2) · cons(n; nil))

tree ::= empty | node(n; t1; t2)

(T1) size(empty) = z

(T2) size(node(n; t1; t2)) = S(size(t1) + size(t2)

(T3) heigh(empty) = z

(T4) height(node(n; t1; t2)) = S(max(heigh(t1), h

(R1) rev(empty) = empty

(R2) rev(node(n; t1; t2)) = node(n; rev(t2); rev(t1))

(12) size(rev(t)) = size(t)

Case(a) t = empty (goal :- size(rev(e

size(rev(empty)) = size(empty) by

Case (b) t = node(n; t1, t2) (goal :- s

size(rev(node(n; t1; t2))) = size(node

= S(si

= S(siz

= node(

heigh(rev(node(n; t1; t2))) = ht(nod

= ht(node(n; rev(t2); rev(t1))) by (re

= S(max(ht(t2), ht(t1)))

= S(max(ht(t1), ht(t2)))

= ht(node(n, t1, t2))

Inorder <left> <root> <right>

pre order <root> <left> <right>

post order <left> <right> <root>

In 1 : In(empty)

In 2 : In(node(n; t

pre 1

pre 2 : pre(n

post 1

post 2 : post(

rev(In(t)) = In(rev(t))

t = node(n, t1, t2) (rev(In(node(n,

rev(In(node(n, t1, t2))) = rev(In(t1) ·

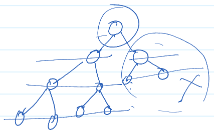
= rev(In(t2))

= In(rev(t2

= In(nod

= In(rev(no

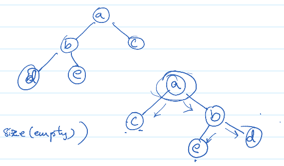
)



$S\left(\max(height(t_1), heigh(t))\right)$

$ight(t_2))$

)



$mpty)) = size(empty))$

rule (R1)

$size(rev(node(n; t_1; t_2)) = size(node(n; t_1, t_2))$

$e(n; rev(t_2); rev(t_1)))$ (R2)

$ze(t_2) + size(t_1)))$ (T2)

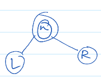
$e(t_1) + size(t_2)))$ ass

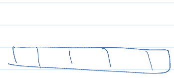
$n; t_1; t_2)$ (T 2)

$le(n; t_1, t_2)$

v 2)

$\max(m, n) = \max(n, m)$





cons(n; nil)

= empty

$_1; t_2) = In(t_1) . (cons(n; nil) . In(t_2))$

$ode(n, t_1, t_2)) = cons(n, nil) . (pre(t_1) . pre(t_2))$

$node(n, t_1, t_2)) = post(t_1) (post(t_2) . cons(n, nil))$

$t_1, t_2))) = In(rev(node(n, t_1, t_2))$

$(cons(n, nil) . In(t_2)))$

rev(cons(n, nil))
= cons(n, nil)

$(rev(cons(n, nil)) . rev(In(t_1)))$

1

→ 14 - 2

$) . cons(n, nil) . In(rev(t_1))$

$e(n, rev(t_2), rev(t_1))$

$le(n, t_1, t_2)))$

(PO1) $post(empty) = nil$

(PO2) $post(node(n;t1;t2)) = post(t1) \cdot (post(t2) \cdot cons(n;nil))$

(15) $rev(In(t)) = In(rev(t))$

induction on t

Case (a) $t = empty$ $\boxed{goal:- rev(In(empty)) = In(rev(empty))}$

$rev(In(empty)) = rev(nil)$ by rule (In1)

$= nil$ by rule (rev1)

$= In(empty)$ by rule (In1)

$= In(rev(empty))$ by rule (rev1)

Case (b) $t = node(n;t1;t2)$

goal:- $rev(In(node(n;t1;t2))) = In(rev(node(n;t1;t2)))$

$rev(In(node(n;t1;t2))) = rev(In(t1) \cdot (cons(n;In(t2))))$ by (In2)

$= rev(cons(n;In(t2))) \cdot rev(In(t1))$ by (rev2)

$= (rev(In(t2)) \cdot rev(cons(n;nil))) \cdot rev(In(t1))$ by (rev2)

$= (In(rev(t2)) \cdot cons(n;nil)) \cdot In(rev(t1))$ by IH

$= In(rev(t2)) \cdot (cons(n;nil) \cdot In(rev(t1)))$ by associativity

$= In(node(n;rev(t2);rev(t1)))$ by (In2)

$= In(rev(node(n;t1;t2)))$ by (rev2)

(16) $\forall t$, tree $t \rightarrow size(t) = |In(t)|$

Induction on t

Case (a): $t = empty$

$size(empty) = Z$ by (t1)

$= |nil|$ by (L1)

$= |In(empty)|$ by (In1)

Case (b): $t = node(n;t1;t2)$

$\boxed{goal: size(node(n;t1;t2)) = |In(node(n;t1;t2))|}$

$size(node(n;t1;t2)) = S(size(t1) + size(t2))$ by (t2)

$= S(|In(t1)| + |In(t2)|)$ by IH

$= |In(t1)| + S(|In(t2)|)$ by (6a) :- $S(n+m) = n + S(m)$

$= |In(t1)| + |cons(n;In(t2))|$ by rule (L2)

$= |In(t1) \cdot (cons(n;In(t2)))|$ by (10c) $|l_1 \cdot l_2| = |l_1| + |l_2|$

$= |In(node(n;t1;t2))|$ by (In2)