Alan V.
15-312 Foundations of Programming Languages
Assignment #1
February 2, 2006

1 Course Mechanics

Task 1
1729

Task 2

1. This is not permitted because you are not allowed to take anything away from a face-to-face meeting (except, presumably, with a TA).
2. This is also not permitted. Online communication requires an hour or so waiting period between the time you discuss the answers and the time you write up the work.
3. I would say that this is also not permitted, because there wasn't enough time between when they consulted and when they wrote up answers. If they had both headed home to write up their answers, however, it would be okay.

2 Associativity of append

Task 1
We can exhibit a derivation:

$$\dfrac{\dfrac{\dfrac{}{\text{zero nat}}}{\text{succ(zero) nat}} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\text{zero nat}}}{\text{succ(zero) nat}}}{\text{succ(succ(zero)) nat}} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\text{zero nat}}}{\text{succ(zero) nat}}}{\text{succ(succ(zero)) nat}}}{\text{succ(succ(succ(zero))) nat}} \quad \dfrac{}{\text{empty list}}}{\text{cons(3,empty) list}}}{\text{append empty cons(3,empty) cons(3,empty)}}}{\text{append cons(2,empty) cons(3,empty) cons(2,cons(3,empty))}}}{\text{append (cons(1,cons(2,empty))) (cons(3,empty)) (cons(1,cons(2,cons(3,empty))))}}$$

Task 2
We will use the induction principle for lists: if *P(empty) and* $\forall h$ *and* $\forall t$ *if* h *nat and P(t) then P(cons(h,t))* then $\forall x$ if x list then P(x). This means we can prove any statement of the form "$\forall x$ if x list then P(x)." We are trying to prove ∀ x if x list then ∀ y if y list then ∃! z ST append x y z. Our P is therefore "∀ y if y list then ∃! z ST append x y z". By the induction principle, TS $\forall x$ if x list then P(x) STS P(empty) and $\forall h$ and $\forall t$ if h nat and P(t) then P(cons(h,t))

1. P(empty): ∀ y if y list then ∃! z ST append empty y z. Set z = y, and $\frac{\text{y list}}{\text{append empty y y}}$, so it does exist. Then, assume some other k ST append empty y k. By inversion, y = k, and so z = k, and so it is unique.

2. TS ∀*h* and ∀*t* if *h* nat and P(*t*) then P(cons(*h*,*t*)). Fix *h* and *t*. We are therefore trying to prove *if h nat and P(t) then P(cons(h,t))*. We can assume the IH of *h* nat and ∀ y if y list then ∃! z ST append *t* y z. This means we simply have to show P(cons(*h*,*t*)). In other words, we have to show ∀ y' if y' list then ∃! z' ST append cons(*h*,*t*) y' z'. Fix y'. We can assume y' list. Set y = y'. Our IH tells us ∃! z ST append *t* y' z. We can apply the rule to get append cons(*h*,*t*) y' cons(*h*,z). Set z' = cons(*h*,z). This proves existence.
   Then assume other k ST append cons(*h*,*t*) y' k. By the rule, k = cons(*h*,k'). We have append cons(*h*,*t*) y' cons(*h*,k'). By inversion, append *t* y' k', but by IH k' = z so k = cons(*h*,z) = z. Therefore, ∃!

Therefore, the mode of append is (∀, ∀, ∃!).

Task 3
We will use the induction principle for append: if ∀ *l if l list then P(empty l l) and* ∀ *h and* ∀ *a b c if h nat and P(a b c) then P(cons(h,a) b cons(h,c))* then *if* ∀ *x y z if append x y z then P(x y z).* This can prove statements of the form "∀ x y z if append x y z then P(x y z)."
We are trying to prove ∀ a b ab if append a b ab then ∀ c abc bc if append ab c abc and append b c bc then append a bc abc. This means our P is "∀ c abc bc if append ab c abc and append b c bc then append a bc abc".

1. ∀ l if l list then P(empty l l). Fix l. if l list then P(empty l l). We can assume l list. We therefore need to show "∀ c abc bc if append l c abc and append l c bc then append empty bc abc." Let's fix c, abc, and bc to leave "if append l c abc and append l c bc then append empty bc abc". We can assume the IH of append l c abc and append l c bc, so we just need to prove append empty bc abc. If we look at our assumptions and our rules, we can conclude that abc = bc since append l c abc and append l c bc for the same l and c, and we proved uniqueness in the previous proof. By the rules then, append empty bc bc.

2. ∀ h and ∀ x y z if h nat and P(x y z) then P(cons(h,x) y cons(h,z)). Let's fix h, x, y, and z. We can assume h nat. We can also assume P(x y z) as our IH: ∀ c abc bc if append z c abc and append y c bc then append x bc abc. However, we must prove the premises before we can use the conclusion; the premises may be false, and the statement would still be true (false ⊃ false is true). We will return to that.
   We are therefore trying to prove ∀ c' abc' bc' if append cons(h,z) c' abc' and append y c' bc' then append cons(h,x) bc' abc'. Let's fix c', abc', and bc'. We can assume append cons(h,z) c' abc' and append y c' bc'. By inversion on append cons(h,z) c' abc', we know that ∃ abc" ST append cons(h,z) c' cons(h,abc"), and append z c' abc". Let's return to our IH. Set c = c' and abc = abc". Set bc = bc'. We have proven the premises for the IH, and so its conclusion must be true, so we can now use the IH.

We are now trying to prove append cons(h,x) bc' cons(h,abc"). We can use inversion on this to reduce it to h nat and append x bc' abc". h nat by assumption. We can prove this statement by the IH by setting bc = bc' and abc = abc". Then, by the rules, append cons(h,x) bc' cons(h,abc").

3 Derived Induction Principles

Task 1
Let's call the P in the "improved" induction principle P' to avoid confusion.
Let's fix a P'. Assume an IH of P'(z), "∀ e if e exp and P'(e) then P(s e)", "∀ e_1 and e_2 if e_1 exp and e_2 exp and P'(e_1) and P'(e_2) then P'(add(e_1,e_2))", and "∀ e_1 and e_2 if e_1 exp and e_2 exp and P'(e_1) and P'(e_2) then P'(mult(e_1,e_2))".
Then we'll fix an e and assume e exp. TS P'(e), we can use rule induction, applying the original inductive principle with P(e) = e exp and P'(e). TS ∀ k exp if k exp then k exp and P'(k) STS

1. P(z): z exp and P'(z). z exp by rules, and P'(z) by IH.
2. ∀ e if e exp and P'(e) then (s e) exp and P'(s e). Fix e. if e exp and P'(e) then (s e) exp and P'(s e). Assume e exp and P'(e) as inner IH. We need to prove (s e) exp and P'(s e). (s e) exp by the rules and IH. P'(s e) by IH since e exp and P'(e).
3. ∀ e_1 and e_2 if e_1 exp and e_2 exp and P'(e_1) and P'(e_2) then add(e_1,e_2) exp and P'(add(e_1,e_2)). Fix e_1 and e_2. if e_1 exp and e_2 exp and P'(e_1) and P'(e_2) then add(e_1,e_2) exp and P'(add(e_1,e_2)) Assume e_1 exp and e_2 exp and P'(e_1) and P'(e_2). We need to show P'(add(e_1,e_2)). By IH, this is true by our assumptions. We can then prove add(e_1,e_2) exp by the rule, since e_1 and e_2 are exp.
4. ∀ e_1 and e_2 if P'(e_1) and P'(e_2) then mult(e_1,e_2) exp and P'(mult(e_1,e_2)). Fix e_1 and e_2. if P'(e_1) and P'(e_2) then mult(e_1,e_2) exp and P'(mult(e_1,e_2)). Assume P'(e_1) and P'(e_2). We need to show P'(mult(e_1,e_2)). By IH, this is true by our assumptions. We can then prove mult(e_1,e_2) exp by the rule, since e_1 and e_2 are exp.

4 Propositional Logic

Task 1

$$\cfrac{\cfrac{\cfrac{}{P_1 thm, P_2 thm \vdash P_1 thm}}{P_1 thm \vdash (P_2 \supset P_1) thm}}{\cdot \vdash P_1 \supset (P_2 \supset P_1)}$$

Task 2
We need to show $value_\rho\ P_1 \supset (P_2 \supset P_1)$ true for all variable assignments ρ. We can simply show a derivation for all possible values of P_1 and P_2.

1. P_1 false. It doesn't matter what P_2 is, since it doesn't matter what B is.

$$\dfrac{\dfrac{value_\rho P_1 false}{value_\rho (P_1 \supset B) true}}{value_\rho (P_1 \supset (P_2 \supset P_1)) true}$$

2. P_1 true, P_2 true.

$$\dfrac{value_\rho P_1 true \qquad \dfrac{value_\rho P_2 true \quad value_\rho P_1 true}{value_\rho (P_2 \supset P_1) true}}{value_\rho (P_1 \supset (P_2 \supset P_1)) true}$$

3. P_1 true, P_2 false.

$$\dfrac{value_\rho P_1 true \qquad \dfrac{value_\rho P_2 false}{value_\rho (P_2 \supset P_1) true}}{value_\rho (P_1 \supset (P_2 \supset P_1)) true}$$

Task 3
We would proceed using the induction principle for theorems. During the proof, we would have to deal with the case of `thm-impl-i`, which requires A thm in the context. Since the context would be empty, we would be unable to deal with this, so the proof would fail.
Let's actually try this using the induction principle for theorems (see below). The statement we would be trying to prove is

> If $\Gamma \vdash A$ thm then $\forall$ variable assignments $\rho\, value_\rho A\, true$

The first case, P($[\Gamma, A\, thm, \Gamma']$,$A$) would be fine. However, it breaks down in the case $\forall \Gamma, A, B$ if P($[\Gamma, A\, thm]$,B) then P($\Gamma, A \supset B$).
Fix Γ, A, B. Assume $\forall$ variable assignments $\rho\, value_\rho A\, true$. Prove $\forall$ variable assignments $\rho\, value_\rho A \supset B\, true$. Fix ρ. Our IH tells us nothing about B, and it's fixed, so we are stuck.

Task 4
We want to prove:

> If $\Gamma \vdash A$ thm then $\forall$ variable assignments ρ, if $\forall X\ ST\, \Gamma = \Gamma', X\, thm, \Gamma''\ value_\rho X true$ then $value_\rho A true$.

and will proceed with the induction principle for theorems: If, *P(*$[\Gamma, A\, thm, \Gamma']$*,A)* and $\forall \Gamma, A, B$ *if P(*$[\Gamma, A\, thm]$*,B) then P(*$\Gamma, A \supset B$*)* and $\forall \Gamma, A, B$ *if P(*$\Gamma, A \supset B$*) and P(*Γ, A*) then P(*Γ, B*)* then $\forall \Gamma, x$ if $\Gamma \vdash x$ then P(Γ,x).
Our P is therefore $\forall$ variable assignments ρ, if $\forall X\ ST\, \Gamma = \Gamma', X\, thm, \Gamma''\ value_\rho X true$ then $value_\rho A true$.

1. P($[\Gamma, A\, thm, \Gamma']$,$A$). In otherwords, $\forall$ variable assignments ρ, if $\forall X\ ST\, \Gamma, A\, thm, \Gamma' = \Gamma'', X\, thm, \Gamma'''\ value_\rho X true$ then $value_\rho A true$. Fix ρ. Assume $\forall X\ ST\, \Gamma, A\, thm, \Gamma' = \Gamma'', X\, thm, \Gamma'''\ value_\rho X true$ as IH. Set X = A, $\Gamma = \Gamma''$ and $\Gamma = \Gamma'''$. By IH, $value_\rho A true$.

2. $\forall \Gamma, A, B$ if P([$\Gamma, A\,thm$],B) then P($\Gamma, A \supset B$). In other words, $\forall \Gamma, A, B$ if $\forall$ variable assignments ρ, if $\forall X\; ST\, \Gamma, A\,thm = \Gamma', X\,thm, \Gamma''\; value_\rho Xtrue$ then $value_\rho Btrue$ then $\forall$ variable assignments ρ', if $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$ then $value_{\rho'} A \supset Btrue$
   Fix Γ, A, B. Assume $\forall$ variable assignments ρ, if $\forall X\; ST\, \Gamma, A\,thm = \Gamma', X\,thm, \Gamma''\; value_\rho Xtrue$ then $value_\rho Btrue$ as IH.
   We want to prove $\forall$ variable assignments ρ', if $\forall X'\, \Gamma = \Gamma'', X'\,thm, \Gamma'''\; value_{\rho'} X'true$ then $value_{\rho'} A \supset Btrue$. Fix ρ'. Assume $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$. We want to prove $value_{\rho'} A \supset Btrue$.
   We first must prove the conditions for the IH. Set $\rho = \rho'$. Set $X = X'$. Set $\Gamma = \Gamma''$ and $\Gamma'' = \Gamma''', Athm$. Our assumption now proves the IH, so we can use it.
   So we want to prove $value_{\rho'}(A \supset B)true$. By IH, $value_\rho Btrue$. We can now look at the logic rules; specifically, `value-impl-f_` and `value-impl-tt`. Since we now know that $value_\rho Btrue$, the entire statement is true for any variable assignment for A.

3. $\forall \Gamma, A, B$ if P($\Gamma, A \supset B$) and P(Γ, A) then P(Γ, B). In other words, $\forall \Gamma, A, B$ if $\forall$ variable assignments ρ, if $\forall X\; ST\, \Gamma = \Gamma', X\,thm, \Gamma''\; value_\rho Xtrue$ then $value_\rho A \supset Btrue$ then $\forall$ variable assignments ρ', if $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$ then $value_{\rho'} Atrue$
   Fix Γ, A, B. Assume the IH of $\forall$ variable assignments ρ, if $\forall X\; ST\, \Gamma = \Gamma', X\,thm, \Gamma''\; value_\rho Xtrue$ then $value_\rho A \supset Btrue$.
   We want to show $\forall$ variable assignments ρ', if $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$ then $value_{\rho'} Atrue$.
   Fix ρ'. Assume $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$. Once again, we must prove the IH. Set $X' = A \supset B$. Set $\rho = \rho'$. Set $\Gamma' = \Gamma'''$ and $\Gamma'' = \Gamma''''$. This proves the IH. So we just need to show $value_{\rho'} Atrue$. We know, by IH, that $value_\rho A \supset Btrue$. Looking at the rules, `value-impl-f_` and `value-impl-tt` both apply. By inversion for `value-impl-tt` , $value_{\rho'} Atrue$. Therefore, we simply need to show the case for `value-impl-f_` could not occur.
   Thankfully, it couldn't, since we have assumed $\forall X'\, \Gamma = \Gamma''', X'\,thm, \Gamma''''\; value_{\rho'} X'true$. Since $\Gamma = \Gamma''''', A, \Gamma''''''$ (since that's true for any X'), and we assumed $value_{\rho'} X'true$ for any X', we know that $value_{\rho'} Atrue$.

Basically, we have proven that, as long as everything contained in Γ is true under any ρ (or, more simply, everything in Γ is a tautology), A is true under any ρ. This implies the case where $\Gamma = \cdot$, since there is nothing contained in $\cdot$, and hence everything in it is true. Note that this does not violate our assumption, since we assumed $\forall X\; ST\, \Gamma = \Gamma', X\,thm, \Gamma''$ and $\cdot$ has no X.