Solution by Xiao Jia

Homework 1

Grading policy for a five-point question:

- Correct: 5
- Minor issues: 4
- Only base case is correct: 2
- Just tried but failed: 1
- Didn't try: 0

Problem 1. (25 points)

$$\frac{}{Z\ nat} \qquad \frac{n\ nat}{S(n)\ nat} \qquad \frac{n\ nat}{add\ Z\ n\ n}\text{AddZ} \qquad \frac{add\ n_1\ n_2\ n_3}{add\ S(n_1)\ n_2\ S(n_3)}\text{AddS}$$

(a) Define $max\ a\ b\ c \iff c = \max(a, b)$

(b) Define $com\ n\ m\ k \iff k = \binom{n}{m}$

(c) Prove that for all n such that $n\ nat$, $add\ n\ Z\ n$ holds

(d) Prove that for all n_1, n_2, n_3, $add\ n_1\ n_2\ n_3 \to add\ n_1\ S(n_2)\ S(n_3)$ holds

(e) Prove that for all n_1, n_2, n_3, $add\ n_1\ n_2\ n_3 \to add\ n_2\ n_1\ n_3$ (addition is commutative)

Solution.

(a)

$$\frac{n\ nat}{max\ Z\ n\ n} \qquad \frac{n\ nat}{max\ S(n)\ Z\ S(n)} \qquad \frac{max\ n_1\ n_2\ n_3}{max\ S(n_1)\ S(n_2)\ S(n_3)}$$

Here is an alternative definition by some of the students:

$$\frac{}{max\ Z\ Z\ Z} \qquad \frac{max\ a\ b\ a}{max\ S(a)\ b\ S(a)}\text{MaxA} \qquad \frac{max\ a\ b\ b}{max\ a\ S(b)\ S(b)}\text{MaxB}$$

This is not a preferable definition, because for any form of $max\ S(x)\ S(y)\ z$, it can be either derived from MaxA or MaxB, so we cannot make use of the admissibility.

(Follow-up question) Prove: If $max\ a\ b\ b$ and $max\ b\ c\ c$, then $max\ a\ c\ c$.

Here is an another definition by GAO Hao 5110309190:

$$\frac{n\ nat}{max\ n\ n\ n} \qquad \frac{n\ nat}{max\ n\ S(n)\ S(n)} \qquad \frac{max\ n_1\ n_2\ n_2 \quad max\ n_2\ n_3\ n_3}{max\ n_1\ n_3\ n_3} \qquad \frac{max\ n_1\ n_2\ n_2}{max\ n_2\ n_1\ n_2}$$

The underlying idea is to define the $\leq$ relation. However, this is not a preferable definition, because it is not based on the structure of max, and the ordering of max-predicates is not trivial. Think about the Ackermann function

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

(b)

$$\frac{n\ nat}{com\ n\ Z\ S(Z)} \qquad \frac{n\ nat}{com\ n\ n\ S(Z)} \qquad \frac{com\ n\ m\ k_1 \quad com\ n\ S(m)\ k_2 \quad add\ k_1\ k_2\ k}{com\ S(n)\ S(m)\ k}$$

because

$$\binom{n+1}{m+1} = \binom{n}{m} + \binom{n}{m+1}$$

Pay attention to the mathematical notation: $k = \binom{n}{m}$, not $\binom{m}{n}$. Many students misused this notation.

This problem actually reflects the Pascal's triangle. Many students missed one rule for $\binom{n}{n}$. Also thanks to HU Xiaojun 5110309201, we have an alternative rule for that:

$$\frac{max\ n\ m\ m}{com\ n\ S(m)\ Z}$$

(c) We prove inductively on the structure of n.

1. Case $\overline{Z\ nat}$: let $n = Z$

$$\begin{aligned} & Z\ nat \\ \text{by AddZ} \vdash\ & add\ Z\ Z\ Z \\ \vdash\ & add\ n\ Z\ n \end{aligned}$$

2. Case $\dfrac{n'\ nat}{S(n')\ nat}$: let $n = S(n')$

$$\begin{aligned} & n'\ nat \\ \text{by I.H.} \vdash\ & add\ n'\ Z\ n' \\ \text{by AddS} \vdash\ & add\ S(n')\ Z\ S(n') \\ \vdash\ & add\ n\ Z\ n \end{aligned}$$

(d) We prove inductively on the structure of $add\ n_1\ n_2\ n_3$.

Pay attention to what our proof is based on. Many students think we are proving inductively on n. That is possible, but needs more work and more knowledge.

1. Case $\dfrac{n\ nat}{add\ Z\ n\ n}$: let $n_1 = Z, n_2 = n_3 = n$

$$\begin{aligned} & n\ nat \\ & \vdash S(n)\ nat \\ \text{by AddZ} & \vdash add\ Z\ S(n)\ S(n) \\ & \vdash add\ n_1\ S(n_2)\ S(n_3) \end{aligned}$$

2. Case $\dfrac{add\ n_1'\ n_2'\ n_3'}{add\ S(n_1')\ n_2'\ S(n_3')}$: let $n_1 = S(n_1'), n_2 = n_2', n_3 = S(n_3')$

$$\begin{aligned} & add\ n_1'\ n_2'\ n_3' \\ \text{by I.H.} & \vdash add\ n_1'\ S(n_2')\ S(n_3') \\ \text{by AddS} & \vdash add\ S(n_1')\ S(n_2')\ S(S(n_3')) \\ & \vdash add\ n_1\ S(n_2)\ S(n_3) \end{aligned}$$

Many students also write $add\ S(n_1')\ n_2'\ S(n_3')$ (by AddS) here. This is true, but not useful. We do not need this fact to prove the final result.

(e) We prove inductively on the structure of $add\ n_1\ n_2\ n_3$.

1. Case $\dfrac{n\ nat}{add\ Z\ n\ n}$: let $n_1 = Z, n_2 = n_3 = n$

$$\begin{aligned} & n\ nat \\ \text{by (c)} & \vdash add\ n\ Z\ n\ nat \\ & \vdash add\ n_2\ n_1\ n_3 \end{aligned}$$

2. Case $\dfrac{add\ n_1'\ n_2'\ n_3'}{add\ S(n_1')\ n_2'\ S(n_3')}$: let $n_1 = S(n_1'), n_2 = n_2', n_3 = S(n_3')$

$$\begin{aligned} & add\ n_1'\ n_2'\ n_3' \\ \text{by I.H.} & \vdash add\ n_2'\ n_1'\ n_3' \\ \text{by (d)} & \vdash add\ n_2'\ S(n_1')\ S(n_3') \\ & \vdash add\ n_2\ n_1\ n_3 \end{aligned}$$

You can reuse previously proved (c) and (d) here.

$\square$

Problem 2. (30 points)

$$\frac{}{\epsilon\ tree} \qquad \frac{t_1\ tree \quad t_2\ tree}{t_1 \oplus t_2\ tree}$$

(a) Define $height\ a\ h \iff a\ tree$ and h is the height of a

(b) Define $size\ a\ n \iff a\ tree$ and n is the number of nodes contained in a

(c) Define $bst\ a \iff a$ is a binary search tree where each node contains a natural number defined in Problem 1. Hint: First define an order on natural numbers.

(d) (15 points) Prove: If $a\ tree$, $height\ a\ h$, and $size\ a\ n$, then $max\ n\ (2^h - 1)\ (2^h - 1)$.

Solution.

(a)

$$\frac{}{height\ \epsilon\ Z} \qquad \frac{height\ t_1\ h_1 \quad height\ t_2\ h_2 \quad max\ h_1\ h_2\ h}{height\ t_1 \oplus t_2\ S(h)}$$

You can reuse previously defined *max* here, or define it again.

(b)

$$\frac{}{size\ \epsilon\ Z} \qquad \frac{size\ t_1\ n_1 \quad size\ t_2\ n_2 \quad add\ n_1\ n_2\ n}{size\ t_1 \oplus t_2\ S(n)}$$

Some of you think the size here is $n_1 + n_2$, which is not true. It is $n_1 + n_2 + 1$; otherwise all trees will have size zero.

(c)

$$\frac{n\ nat}{less\ Z\ S(n)} \qquad \frac{less\ n_1\ n_2}{less\ S(n_1)\ S(n_2)}$$

Many students define a less-than-or-equal-to ($\leq$) relation here. This is wrong. A binary search tree, by definition, contains *unique* keys in its nodes; otherwise when the keys are identical, you don't know which value is the one you want.

$$\frac{n\ nat}{maxval\ \texttt{leaf}(n)\ n} \qquad \frac{maxval\ t_2\ m}{maxval\ \texttt{node}(n; t_1; t_2)\ m}$$

$$\frac{n\ nat}{minval\ \texttt{leaf}(n)\ n} \qquad \frac{minval\ t_1\ m}{minval\ \texttt{node}(n; t_1; t_2)\ m}$$

Many students omit this part, or just describe them in English words, while you are actually required to define them mathematically.

$$\frac{n\ nat}{bst\ \texttt{leaf}(n)} \qquad \frac{bst\ t_1 \quad bst\ t_2 \quad maxval\ t_1\ l \quad minval\ t_2\ r \quad less\ l\ n \quad less\ n\ r}{bst\ \texttt{node}(n; t_1; t_2)}$$

Some of you only compare the values on the *roots*, which is not a guarantee for a BST.

(d) This question is actually not valid, because the *syntax* $2^h - 1$ is undefined. So in order to solve it correctly, we have to define the meaning of this syntax. Then the key idea of the proof is that $max\ a\ b\ b \iff a \leq b$. Most students didn't come up with a solution at all, so it's left to you to solve it again after this tutorial (where you can hopefully gain more knowledge on formal definitions and proofs).

$\square$

Problem 3. (25 points)

$$\frac{}{\epsilon\ list} \qquad \frac{n\ nat \quad l\ list}{n :: l\ list}$$

(a) Define $append\ n\ l\ l' \iff l'$ is l appended with natural number n

(b) Define $reverse\ l\ l' \iff l'$ is the reverse of list l

(c) Define $sum\ l\ n \iff$ the sum of all elements in list l is n

(d) (10 points) Prove: If $sum\ l\ n$, and $reverse\ l\ l'$, then $sum\ l'\ n$.

Solution.

(a)

$$\frac{n\ nat}{append\ n\ \epsilon\ (n :: \epsilon)}\ \text{AppZ} \qquad \frac{append\ n\ l\ l' \quad n'\ nat}{append\ n\ (n' :: l)\ (n' :: l')}\ \text{AppN}$$

Many students misused the notations of *append* and ::. The form of *append* is *append n list list*, not *append list n list*. The form of :: is $n :: list$, not $list :: n$ (so $\epsilon :: n$ is wrong). Some of you also missed the $n\ nat$ here.

(b)

$$\frac{}{reverse\ \epsilon\ \epsilon}\ \text{RevZ} \qquad \frac{reverse\ l\ l' \quad append\ n\ l'\ l''}{reverse\ (n :: l)\ l''}\ \text{RevN}$$

(c)

$$\frac{}{sum\ \epsilon\ Z}\ \text{SumZ} \qquad \frac{sum\ l\ n' \quad add\ n\ n'\ n''}{sum\ (n :: l)\ n''}\ \text{SumN}$$

(d) Below is a structure of the solution. Many students didn't come up with a solution at all, so you can try it again after this tutorial.

First we prove the following lemma. You may need to prove more lemmas to prove this lemma.

Lemma AS: $\dfrac{append\ n\ l\ l' \quad sum\ l\ m \quad add\ n\ m\ m'}{sum\ l'\ m'}$

Then we prove inductively on the structure of *sum*.

1. Case $\dfrac{}{sum\ \epsilon\ Z}$
   (1) Let $l = \epsilon, n = Z$ (by assumption)
   (2) $reverse\ l\ l'$ (by assumption)
   (3) $l' = \epsilon$ (by RevZ)
   (4) $sum\ l'\ n$ (by SumZ)

2. Case $\dfrac{sum\ l_1\ n_2 \quad add\ n_1\ n_2\ n}{sum\ (n_1 :: l_1)\ n}$

   (1) Let $l = n :: l_1$ (by assumption)

   (2) $\dfrac{reverse\ l_1\ l_2 \quad append\ n_1\ l_2\ l'}{reverse\ (n_1 :: l_1)\ l'}$ (by RevN)

   (3) $sum\ l_1\ n_2$ (by asumption)

   (4) $reverse\ l_1\ l_2$ (by (2))

   (5) $sum\ l_2\ n_2$ (by I.H., (3), and (4))

   (6) $append\ n_1\ l_2\ l'$ (by (2))

   (7) $add\ n_1\ n_2\ n$ (by assumption)

   (8) $sum\ l'\ n$ (by Lemma AS, (5), (6), and (7))

$\square$

Problem 4. Here is a definition of the less-than-or-equal-to judgement for natural numbers:
Judgement Form: $\vdash leq\ n_1\ n_2$
Rules:

$$\frac{n_2\ nat}{\vdash leq\ Z\ n_2} \qquad \text{(Z-Leq)}$$

$$\frac{\vdash leq\ n_1\ n_2}{\vdash leq\ (S\ n_1)\ (S\ n_2)} \qquad \text{(S-Leq)}$$

(a) (10 points) Use the *leq* judgement to define a new judgement with the form

$$\vdash ascend\ l$$

that is valid whenever the elements of l are in ascending order (duplicates are allowed). For example, these judgements are valid:

$$\vdash ascend\ cons(Z, cons(Z, cons(S\ S\ S\ Z, cons(S\ S\ S\ S\ S\ S\ Z, nil))))$$

$$\vdash ascend\ nil$$

$$\vdash ascend\ cons(S\ S\ Z, nil)$$

This judgement is not valid:

$$\vdash ascend\ cons(Z, cons(S\ S\ Z, cons(S\ Z, nil)))$$

(b) (10 points) Consider the judgement $\vdash dup\ l_1\ l_2$ and its rules:

$$\frac{}{\vdash dup\ nil\ nil} \qquad \text{(Nil-Dup)}$$

$$\frac{\vdash dup\ l_1\ l_2}{\vdash dup\ cons(n, l_1)\ cons(n, cons(n, l_2))} \qquad \text{(Cons-Dup)}$$

Prove: If $\vdash ascend\ l_1$ and $\vdash dup\ l_1\ l_2$ then $\vdash ascend\ l_2$.

Solution.

(a)

$$\frac{}{\vdash \textit{ascend nil}} \text{ (3 points)} \tag{Asc0}$$

$$\frac{n \textit{ nat}}{\vdash \textit{ascend cons}(n, \textit{nil})} \text{ (3 points)} \tag{Asc1}$$

$$\frac{\vdash \textit{leq } n_1 \ n_2 \quad \vdash \textit{ascend cons}(n_2, l)}{\vdash \textit{ascend cons}(n_1, \textit{cons}(n_2, l))} \text{ (4 points)} \tag{Asc2}$$

Many students didn't consider the Asc1 case, where the list only contains one element. Minor issues in this problem only cost 1 point.

(b) First we prove the following lemma.

Lemma L: $\dfrac{n \textit{ nat}}{\vdash \textit{leq } n \ n}$

1. Case $\dfrac{}{Z \textit{ nat}}$
   (1) $\vdash \textit{leq } Z \ Z$ (by Z-Leq)
2. Case $\dfrac{n \textit{ nat}}{S(n) \textit{ nat}}$
   (1) $\vdash \textit{leq } n \ n$ (by I.H.)
   (2) $\vdash \textit{leq } S(n) \ S(n)$ (by S-Leq)

Then we prove inductively on the structure of $\vdash \textit{ascend } l$.

1. Case $\dfrac{}{\vdash \textit{ascend nil}}$
   (1) Let $l_1 = \textit{nil}$
   (2) $\vdash \textit{dup nil nil}$ (by Nil-Dup, and assumption)
   (3) Let $l_2 = \textit{nil}$
   (4) $\vdash \textit{ascend } l_2$ (by Asc0)
2. Case $\dfrac{n \textit{ nat}}{\vdash \textit{ascend cons}(n, \textit{nil})}$
   (1) Let $l_1 = \textit{cons}(n, \textit{nil})$
   (2) $\vdash \textit{dup nil nil}$ (by Nil-Dup)
   (3) $\vdash \textit{dup cons}(n, \textit{nil}) \ \textit{cons}(n, \textit{cons}(n, \textit{nil}))$ (by Cons-Dup, and (2))
   (4) Let $l_2 = \textit{cons}(n, \textit{cons}(n, \textit{nil}))$
   (5) $\vdash \textit{ascend cons}(n, \textit{nil})$ (by Asc1)
   (6) $\vdash \textit{leq } n \ n$ (by Lemma L)
   (7) $\vdash \textit{ascend } l_2$ (by Asc2, (5), and (6))
3. Case $\dfrac{\vdash \textit{leq } n_1 \ n_2 \quad \vdash \textit{ascend cons}(n_2, l)}{\vdash \textit{ascend cons}(n_1, \textit{cons}(n_2, l))}$

(1) Let $l_1 = cons(n_1, cons(n_2, l)), l_3 = cons(n_2, l)$

(2) $\vdash dup\ l_3\ cons(n_2, cons(n_2, l'))$ and let $l_4 = cons(n_2, cons(n_2, l'))$ so $\vdash dup\ l_3\ l_4$

(3) $\vdash dup\ l_1\ cons(n_1, cons(n_1, l_4))$

(4) let $l_2 = cons(n_1, cosn(n_1, l_4))$

(5) $\vdash ascend\ l_3$ (by assumption)

(6) $\vdash ascend\ l_4$ (by I.H., (2), and (5))

(7) $\vdash ascend\ cons(n_2, cons(n_2, l'))$

(8) $\vdash leq\ n_1\ n_2$ (by assumption)

(9) $\vdash ascend\ cons(n_1, cons(n_2, (cons(n_2, l'))))$ (by Asc2, (8), and (7))

(10) $\vdash ascend\ cons(n_1, l_4)$

(11) $\vdash leq\ n_1\ n_1$ (by Lemma L)

(12) $\vdash ascend\ l_2$ (by Asc2, (4), (11), and (10))

$\square$