In-Class Problems Week 7, Wed.

Problem 1.
The Elementary 18.01 Functions (F18's) are the set of functions of one real variable defined recursively as follows:

Base cases:

- The identity function, $\text{id}(x) ::= x$ is an F18,
- any constant function is an F18,
- the sine function is an F18,

Constructor cases:

If f, g are F18's, then so are

1. $f + g$, fg, e^g (the constant e),
2. the inverse function $f^{(-1)}$,
3. the composition $f \circ g$.

(a) Prove that the function $1/x$ is an F18.

Warning: Don't confuse $1/x = x^{-1}$ with the inverse, $\text{id}^{(-1)}$ of the identity function $\text{id}(x)$. The inverse $\text{id}^{(-1)}$ is equal to id.

(b) Prove by Structural Induction on this definition that the Elementary 18.01 Functions are *closed under taking derivatives*. That is, show that if $f(x)$ is an F18, then so is $f' ::= df/dx$. (Just work out 2 or 3 of the most interesting constructor cases; you may skip the less interesting ones.)

Problem 2.
Let p be the string [ ] . A string of brackets is said to be *erasable* iff it can be reduced to the empty string by repeatedly erasing occurrences of p. For example, here's how to erase the string [[[ ]] [ ]] [ ]:

$$[\,[\,[\,]\,]\,[\,]\,]\,[\,] \rightarrow [\,[\,]\,] \rightarrow [\,] \rightarrow \lambda.$$

On the other hand the string [ ]] [[[[[ ]] is not erasable because when we try to erase, we get stuck:

$$[\,]\,]\,[\,[\,[\,[\,[\,]\,] \rightarrow \,]\,[\,[\,[\,[\, \rightarrow \,]\,[\,[\,[\, \not\rightarrow$$

Let Erasable be the set of erasable strings of brackets. Let RecMatch be the recursive data type of strings of *matched* brackets given in Definition 11.3.7.

(a) Use structural induction to prove that

$$\text{RecMatch} \subseteq \text{Erasable}\,.$$

(b) Supply the missing parts of the following proof that

$$\text{Erasable} \subseteq \text{RecMatch}\,.$$

Proof. We prove by induction on the length, n, of strings, x, that if $x \in \text{Erasable}$, then $x \in \text{RecMatch}$. The induction predicate is

$$P(n) ::= \forall x \in \text{Erasable}\,.\,(|x| \leq n \quad \text{IMPLIES} \quad x \in \text{RecMatch})$$

Base case:

What is the base case? Prove that P is true in this case.

Inductive step: To prove $P(n+1)$, suppose $|x| \leq n+1$ and $x \in \text{Erasable}$. We need only show that $x \in \text{RecMatch}$. Now if $|x| < n+1$, then the induction hypothesis, $P(n)$, implies that $x \in \text{RecMatch}$, so we only have to deal with x of length exactly $n+1$.

Let's say that a string y is an *erase* of a string z iff y is the result of erasing a single occurrence of p in z.

Since $x \in \text{Erasable}$ and has positive length, there must be an erase, $y \in \text{Erasable}$, of x. So $|y| = n-1$, and since $y \in \text{Erasable}$, we may assume by induction hypothesis that $y \in \text{RecMatch}$.

Now we argue by cases:

Case (y is the empty string).

Prove that $x \in \text{RecMatch}$ in this case.

Case ($y = \mathbf{[}\ s\ \mathbf{]}\ t$ for some strings $s, t \in \text{RecMatch}$.) Now we argue by subcases.

- **Subcase** (x is of the form $\mathbf{[}\ s'\ \mathbf{]}\ t$ where s is an erase of s').
  Since $s \in \text{RecMatch}$, it is erasable by part (b), which implies that $s' \in \text{Erasable}$. But $|s'| < |x|$, so by induction hypothesis, we may assume that $s' \in \text{RecMatch}$. This shows that x is the result of the constructor step of RecMatch, and therefore $x \in \text{RecMatch}$.
- **Subcase** (x is of the form $\mathbf{[}\ s\ \mathbf{]}\ t'$ where t is an erase of t').
  Prove that $x \in \text{RecMatch}$ in this subcase.
- **Subcase**($x = p\mathbf{[}\ s\ \mathbf{]}\ t$).
  Prove that $x \in \text{RecMatch}$ in this subcase.

The proofs of the remaining subcases are just like this last one. **List these remaining subcases.**

This completes the proof by induction on n, so we conclude that $P(n)$ holds for all $n \in \mathbb{N}$. Therefore $x \in \text{RecMatch}$ for every string $x \in \text{Erasable}$. That is,

$$\text{Erasable} \subseteq \text{RecMatch} \text{ and hence } \text{Erasable} = \text{RecMatch}\,.$$

■

Problem 3.
Here is a simple recursive definition of the set, E, of even integers:

Definition. Base case: $0 \in E$.

Constructor cases: If $n \in E$, then so are $n + 2$ and $-n$.

Provide similar simple recursive definitions of the following sets:

(a) The set $S ::= \left\{2^k 3^m 5^n \mid k, m, n \in \mathbb{N}\right\}$.

(b) The set $T ::= \left\{2^k 3^{2k+m} 5^{m+n} \mid k, m, n \in \mathbb{N}\right\}$.

(c) The set $L ::= \left\{(a, b) \in \mathbb{Z}^2 \mid 3 \mid (a - b)\right\}$.

Let L' be the set defined by the recursive definition you gave for L in the previous part. Now if you did it right, then $L' = L$, but maybe you made a mistake. So let's check that you got the definition right.

(d) Prove by structural induction on your definition of L' that

$$L' \subseteq L.$$

(e) Confirm that you got the definition right by proving that

$$L \subseteq L'.$$

(f) See if you can give an *unambiguous* recursive definition of L.

6.042J / 18.062J Mathematics for Computer Science
Spring 2010