CSCI 5535: Homework Assignment 0: Preliminaries

Fall 2023: Due Friday, September 8, 2023

The purpose of this assignment is to refresh preliminaries from prior courses. It is all right if you find this assignment difficult. Start early and ask for help if you get stuck! In particular, you are encouraged to ask questions in class, on the discussion forum, or in office hours (though do *not* post your solutions directly). You are also welcome to talk about these questions in larger groups. However, be sure to acknowledge those with which you discussed.

Recall the evaluation guideline from the course syllabus.

> *Both your ideas and also the clarity with which they are expressed matter—both in your English prose and your code!*
>
> *We will consider the following criteria in our grading:*
>
> - How well does your submission answer the questions? *For example, a common mistake is to give an example when a question asks for an explanation. An example may be useful in your explanation, but it should not take the place of the explanation.*
> - How clear is your submission? *If we cannot understand what you are trying to say, then we cannot give you points for it. Try reading your answer aloud to yourself or a friend; this technique is often a great way to identify holes in your reasoning. For code, not every program that "works" deserves full credit. We must be able to read and understand your intent. Make sure you state any preconditions or invariants for your functions.*

Submission Instructions. Typesetting is preferred but scanned, clearly legible handwritten write-ups are acceptable. Please no other formats—no `.doc` or `.docx`. You may use whatever tool you wish (e.g., LaTeX, Word, markdown, plain text, pencil+paper) as long as it is legibly converted into a `pdf`.

1. **Feedback**. Complete the survey on the linked from Canvas after completing this assignment. Any non-empty answer will receive full credit.

2. **Course Mechanics**. The purpose of this question is to ensure that you get familiar with this course's collaboration policy and the reasoning behind it. As in any class, you are responsible for following our collaboration policy.

   Our course's collaboration policy is in the course syllabus. Read it; then, for each of the following situations, decide whether or not the students' actions are permitted by the policy. Briefly explain your answers.

(a) Dolores and Toby are discussing Problem 3 by IM. Meanwhile, Toby is writing up his solution to that problem.

(b) Amy, Jeff, and Chris split a pizza while talking about their homework, and by the end of lunch, their pizza box is covered with notes and solutions. Chris throws out the pizza box and the three go to class.

(c) Ian and Jeremy write out a solution to Problem 4 on a whiteboard in CSEL. Then, they erase the whiteboard and run to the lobby. Sitting at separate tables, each student types up the solution on his laptop.

(d) Nitin and Margaret are working on this homework over lunch; they write out a solution to Problem 2 on a napkin. After lunch, Nitin pockets the napkin, heads home, and writes up his solution.

3. **Set Theory Preliminaries**. Let X and Y be sets. Let $\wp(X)$ denote the powerset of X (i.e., the set of all subsets of X). There is a 1-1 correspondence (i.e., a bijection) between the sets A and B where

$$A \stackrel{\text{def}}{=} X \to \wp(Y) \qquad \text{and} \qquad B \stackrel{\text{def}}{=} \wp(X \times Y)\,.$$

Note that A is a set of functions and B is a (or can be viewed as a) set of relations. This correspondence will allow us to use functional notation for certain sets in class.

Demonstrate the correspondence between A and B by presenting an appropriate function and proving that it is a bijection. Hint: you might construct a function $f : B \to A$ and prove that f is an injection and a surjection.

This exercise is Exercise 1.4 from page 8 of Winskel's book.

4. **Induction Fallacy**. Find the flaw in the following inductive proof that "All flowers smell the same." Please indicate exactly which sentences are wrong in the proof. Giving a counterexample does not constitute an acceptable solution.

Proof. Let F be the set of all flowers, and let $\mathsf{smells}(f)$ be the smell of the flower $f \in F$ (the range of smells is not so important, but we will assume that it admits equality). We will also assume that F is countable. Let the property $P(n)$ mean that all subsets of F of size at most n contain flowers that smell the same.

$$P(n) \stackrel{\text{def}}{=} \forall X \in \wp(F).\, |X| \le n \implies (\forall f, f' \in X.\, \mathsf{smells}(f) = \mathsf{smells}(f'))$$

The notation $|X|$ denotes the number of elements of X.

One way to formulate the statement to prove is $\forall n \ge 1. P(n)$. We will prove this by induction on n, as follows:

Case (Base Case: $n = 1$). Obviously all singleton sets of flowers contain flowers that smell the same (by the definition of $P(n)$).

Case (Induction Step). Let n be arbitrary, and assume that all subsets of F of size at most n contain flowers that smell the same. We will prove that the same thing holds for all subsets of size at most $n+1$. Pick an arbitrary set X such that $|X| = n+1$. Pick two distinct flowers $f, f' \in X$, and let's show that $\mathsf{smells}(f) = \mathsf{smells}(f')$. Let $Y = X - \{f\}$ and $Y' = X - \{f'\}$. Obviously, Y and Y' are sets of size at most n, so the induction hypothesis holds for both of them.

Pick any arbitrary $x \in Y \cap Y'$. Obviously, $x \neq f$ and $x \neq f'$. We have that $\mathsf{smells}(f') = \mathsf{smells}(x)$ (from the induction hypothesis on Y) and $\mathsf{smells}(f) = \mathsf{smells}(x)$ (from the induction hypothesis on Y'). Hence $\mathsf{smells}(f) = \mathsf{smells}(f')$, which proves the inductive step and the theorem.

□

> One indication that the proof might be wrong is the large number of occurrences of the word "obviously" `:-)`.

5. **Induction Preliminaries**. Consider the abstract syntax of the following arithmetic expression language defined inductively as follows:

$$\begin{array}{lrcl} \text{expressions} & e \in \textbf{Expr} & ::= & n \mid -e_1 \mid e_1 + e_2 \\ \text{integers} & n \in \mathbb{Z} & & \end{array}$$

Prove by induction the following statement:

> For all expressions $e \in \textbf{Expr}$, there is at least one integer constant n.

It is not difficult to see that the above statement is true. The exercise is to construct a crisp mathematical argument, so especially on this first exercise, err on the side of being pedantic.

6. **Judgments: Shuffling Cards**. For this question, we will play with cards. Rather than the standard 52 different cards, we will define four different cards, one for each suit. We model a deck of cards as a list.

$$\frac{}{\heartsuit\ \text{card}} \qquad \frac{}{\spadesuit\ \text{card}} \qquad \frac{}{\clubsuit\ \text{card}} \qquad \frac{}{\diamondsuit\ \text{card}} \qquad \frac{}{\mathsf{nil}\ \text{deck}} \qquad \frac{c\ \text{card} \quad s\ \text{deck}}{\mathsf{cons}(c, s)\ \text{deck}}$$

These rules are an iterated inductive definition for a deck of cards.

We also want to define a judgment form $\mathsf{unshuffle}(s_1, s_2, s_3)$. Shuffling takes two decks of cards and creates a new deck of cards by interleaving the two decks in some way; *un*-shuffling is just the opposite operation.

The definition of $\mathsf{unshuffle}(s_1, s_2, s_3)$ defines a relation between three decks of cards s_1, s_2, and s_3, where s_2 and s_3 are arbitrary "unshufflings" of the first deck—sub-decks where the order from the original deck is preserved so that the two sub-decks s_2 and s_3 could potentially be shuffled back to produce the original deck s_1.

$$\frac{}{\mathsf{unshuffle}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})} \qquad \frac{c\ \text{card} \quad \mathsf{unshuffle}(s_1, s_2, s_3)}{\mathsf{unshuffle}(\mathsf{cons}(c, s_1), s_2, \mathsf{cons}(c, s_3))}$$

$$\frac{c\ \text{card} \quad \mathsf{unshuffle}(s_1, s_2, s_3)}{\mathsf{unshuffle}(\mathsf{cons}(c, s_1), \mathsf{cons}(c, s_2), s_3)}$$

(a) Prove the following judgment. There are at least two ways to do so.

$$\mathsf{unshuffle}(\mathsf{cons}(\heartsuit, \mathsf{cons}(\spadesuit, \mathsf{cons}(\spadesuit, \mathsf{cons}(\diamondsuit, \mathsf{nil})))), \mathsf{cons}(\spadesuit, \mathsf{cons}(\diamondsuit, \mathsf{nil})), \mathsf{cons}(\heartsuit, \mathsf{cons}(\spadesuit, \mathsf{nil})))$$

(b) Give an inductive definition defining $\mathsf{separate}(s_1, s_2, s_3)$, a judgment form similar to $\mathsf{unshuffle}(s_1, s_2, s_3)$ that relates a deck of cards to two sub-decks where all of the red cards (suits $\diamondsuit$ and $\heartsuit$) are in one deck and all the black cards (suits $\clubsuit$ and $\spadesuit$) are in the other. The following should be provable from your inductive definition:

$$\mathsf{separate}(\mathsf{cons}(\heartsuit, \mathsf{cons}(\diamondsuit, \mathsf{cons}(\spadesuit, \mathsf{nil}))), \mathsf{cons}(\heartsuit, \mathsf{cons}(\diamondsuit, \mathsf{nil})), \mathsf{cons}(\spadesuit, \mathsf{nil}))$$
$$\mathsf{separate}(\mathsf{cons}(\spadesuit, \mathsf{cons}(\diamondsuit, \mathsf{cons}(\clubsuit, \mathsf{cons}(\heartsuit, \mathsf{nil})))), \mathsf{cons}(\diamondsuit, \mathsf{cons}(\heartsuit, \mathsf{nil})), \mathsf{cons}(\spadesuit, \mathsf{cons}(\clubsuit, \mathsf{nil})))$$
$$\mathsf{separate}(\mathsf{cons}(\clubsuit, \mathsf{cons}(\heartsuit, \mathsf{cons}(\clubsuit, \mathsf{cons}(\spadesuit, \mathsf{nil})))), \mathsf{cons}(\heartsuit, \mathsf{nil}), \mathsf{cons}(\clubsuit, \mathsf{cons}(\clubsuit, \mathsf{cons}(\spadesuit, \mathsf{nil}))))$$

However $\mathsf{separate}(\mathsf{cons}(\heartsuit, \mathsf{cons}(\spadesuit, \mathsf{nil})), \mathsf{cons}(\heartsuit, \mathsf{cons}(\spadesuit, \mathsf{nil})), \mathsf{nil})$ should **not** be provable from your definition, because the deck in the second position has both a red and a black card.

Similarly, $\mathsf{separate}(\mathsf{cons}(\heartsuit, \mathsf{cons}(\diamondsuit, \mathsf{nil})), \mathsf{cons}(\diamondsuit, \mathsf{cons}(\heartsuit, \mathsf{nil})), \mathsf{nil})$ should not be provable from your definitions, because ordering is not preserved.