

# Assignment 0: Rule Induction

15-312: Principles of Programming Languages

Out: Tuesday, January 15th, 2013

Due: Tuesday, January 22nd, 2013, 11:59PM

Welcome to 15-312! First things first. We will be using Piazza for all class communications. If you have already received a welcome e-mail, there is nothing more you need to do. If not, please subscribe post-haste at <http://piazza.com/class#spring2013/15312>.

Go to the course web page to understand the whiteboard policy for collaboration regarding the homework assignments, the late policy regarding timeliness of homework submissions, and the use of Piazza.

Homework will typically consist of a theoretical section and an implementation section. For the first assignment, there is only the theoretical section. You are required to typeset your answers; see the course Web page for some guidance.

In this first assignment we are asking you to practice proving theorems by rule induction. You may find this assignment difficult. Start early, and ask us for help if you get stuck! In particular, you are encouraged to ask the TAs for help over Piazza, and/or come to office hours.

## Submission

To submit your solutions place a file named `assn0.pdf` in your handin directory:

`/afs/andrew.cmu.edu/course/15/312/handin/<yourandrewid>/assn0/`

## 1 Course Mechanics

The purpose of this question is to ensure that you get familiar with this course's collaboration policy.

As in any class, you are responsible for following our collaboration policy; violations will be handled according to university policy.

**Task 1.1** (4 pts). Our course's collaboration policy is on the course's Web site. Read it; then, for each of the following situations, decide whether or not the students' actions are permitted by the policy. Explain your answers.

1. Dolores and Toby are discussing Problem 3 by IM. Meanwhile, Toby is writing up his solution to that problem.
2. 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.

3. Ian and Jeremy write out a solution to Problem 4 on a whiteboard in Newell-Simon Hall. Then, they erase the whiteboard and run to the atrium. Sitting at separate tables, each student types up the solution on his laptop.
4. 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.

## 2 Shuffling cards

For this assignment, 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 stack of cards as a list (don't confuse a stack of cards with the data structure of stacks).

$$\overline{\heartsuit \text{ card}} \quad (1) \quad \overline{\spadesuit \text{ card}} \quad (2) \quad \overline{\clubsuit \text{ card}} \quad (3) \quad \overline{\diamondsuit \text{ card}} \quad (4)$$

$$\overline{\text{nil stack}} \quad (5) \quad \frac{c \text{ card} \quad s \text{ stack}}{\text{cons}(c, s) \text{ stack}} \quad (6)$$

These rules are an iterated inductive definition for a stack of cards; these rules lead to the following induction principle:

In order to show  $\mathcal{P}(s)$  whenever  $s$  stack, it is enough to show

1.  $\mathcal{P}(\text{nil})$
2.  $\mathcal{P}(\text{cons}(c, s))$  assuming  $c$  card and  $\mathcal{P}(s)$

We also want to define an judgment unshuffle. Shuffling takes two stacks of cards and creates a new stack of cards by interleaving the two stacks in some way; un-shuffling is just the opposite operation.

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

$$\overline{\text{unshuffle}(\text{nil}, \text{nil}, \text{nil})} \quad (7) \quad \frac{c \text{ card} \quad \text{unshuffle}(s_1, s_2, s_3)}{\text{unshuffle}(\text{cons}(c, s_1), s_2, \text{cons}(c, s_3))} \quad (8)$$

$$\frac{c \text{ card} \quad \text{unshuffle}(s_1, s_2, s_3)}{\text{unshuffle}(\text{cons}(c, s_1), \text{cons}(c, s_2), s_3)} \quad (9)$$

**Task 2.1** (10 pts). Prove the following (by giving a derivation). There are at least two ways to do so.

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

**Task 2.2** (5 pts). What was the other way? (describe briefly, or just give the other derivation)

**Task 2.3** (15 pts). Prove that unshuffle has mode  $(\forall, \exists, \exists)$ . That is, prove the following:

For all  $s_1$ , if  $s_1$  stack, then there exists  $s_2$  and  $s_3$  such that  $\text{unshuffle}(s_1, s_2, s_3)$ .

Note that there are a number of different ways of proving this! What the  $s_2$  and  $s_3$  “look like” may be very different depending on how you write the proof. Restate any induction principle you use, and identify what property  $P$  you are proving with that induction principle.

**Task 2.4** (15 pts). Give an inductive definition of separate, a judgment similar to unshuffle that relates a stack of cards to two “un-shuffled” sub stacks where all of the red cards (suits  $\diamond$  and  $\heartsuit$ ) are in one stack and all the black cards (suits  $\clubsuit$  and  $\spadesuit$ ) are in the other. The following should be provable from your inductive definition:

$\text{separate}(\text{cons}(\heartsuit, \text{cons}(\diamond, \text{cons}(\spadesuit, \text{nil}))), \text{cons}(\heartsuit, \text{cons}(\diamond, \text{nil})), \text{cons}(\spadesuit, \text{nil}))$   
 $\text{separate}(\text{cons}(\spadesuit, \text{cons}(\diamond, \text{cons}(\clubsuit, \text{cons}(\heartsuit, \text{nil})))), \text{cons}(\diamond, \text{cons}(\heartsuit, \text{nil})), \text{cons}(\spadesuit, \text{cons}(\clubsuit, \text{nil})))$   
 $\text{separate}(\text{cons}(\clubsuit, \text{cons}(\heartsuit, \text{cons}(\clubsuit, \text{cons}(\spadesuit, \text{nil})))), \text{cons}(\heartsuit, \text{nil}), \text{cons}(\clubsuit, \text{cons}(\clubsuit, \text{cons}(\spadesuit, \text{nil}))))$

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

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

**Task 2.5** (5 pts). Hopefully, your definition of separate will have not just the mode  $(\forall, \exists, \exists)$ , but the stronger mode  $(\forall, \exists!, \exists!)$ . What does this mode mean? Why does unshuffle *not* have this mode?

### 3 Cutting cards

For this part of the assignment we will define, using simultaneous inductive definition, stacks of cards with even or odd numbers of cards in them.

$$\frac{}{\text{nil even}} \text{ (10)} \quad \frac{c \text{ card} \quad s \text{ odd}}{\text{cons}(c, s) \text{ even}} \text{ (11)} \quad \frac{c \text{ card} \quad s \text{ even}}{\text{cons}(c, s) \text{ odd}} \text{ (12)}$$

This inductive definition is *simultaneous* (because it simultaneously defines **even** and **odd**) as well as *iterated* (because it relies on the previously-defined definition of **card**).

**Task 3.1** (6 pts). What is the induction principle for these judgments? You may want to examine the induction principle for **even** and **odd** natural numbers from PFPL.

**Task 3.2** (15 pts). Prove well-formedness for the **even** judgment. That is, prove “For all  $s$ , if  $s$  even then  $s$  stack.”

You should use the induction principle from the previous task. Again, be sure to identify what property or properties you are proving with that induction principle.

**Task 3.3** (10 pts). Prove the following theorem:

For all  $S$ , if

1.  $S(\text{nil})$ .
2. For all  $c_1$ ,  $c_2$ , and  $s$ , if  $c_1$  card,  $c_2$  card, and  $S(s)$ , then  $S(\text{cons}(c_1, \text{cons}(c_2, s)))$ .

then for all  $s$ , if  $s$  even then  $S(s)$ .

You will want to use the induction principle mentioned above in order to prove this; as always, remember to carefully consider and state the induction hypothesis you are using.

Note: this is a difficult proof, because the induction hypothesis is not immediately obvious. Here's a hint: because you are dealing with a simultaneous inductive definition, the induction hypothesis will have two parts. In our solution, the induction hypothesis pertaining to even-sized stacks is " $S(s)$ ," and the one pertaining to odd-size stacks is "For all  $c'$ , if  $c'$  card then  $S(\text{cons}(c', s))$ ."

Proving this statement justifies a new induction principle, a *derived induction principle*:

To show that  $S(s)$  whenever  $s$  even, it is enough to show

- $S(\text{nil})$
- $S(\text{cons}(c_1, \text{cons}(c_2, s)))$ , assuming  $c_1$  card,  $c_2$  card, and  $S(s)$

**Task 3.4** (15 pts). Another "operation" on cards is *cutting*, where a player separates a single stack of cards into two stacks of cards by removing some number of cards from the top of the stack. We can define cutting cards using an inductive definition.

$$\frac{s \text{ stack}}{\text{cut}(s, s, \text{nil})} \quad (13) \qquad \frac{c \text{ card} \quad \text{cut}(s_1, s_2, s_3)}{\text{cut}(\text{cons}(c, s_1), s_2, \text{cons}(c, s_3))} \quad (14)$$

Using the derived induction principle from the previous task (you can use the induction principle from the previous task even if you do not do the previous task!), prove the following:

For all  $s_1$ ,  $s_2$ ,  $s_3$ , if  $s_2$  even,  $s_3$  even, and  $\text{cut}(s_1, s_2, s_3)$ , then  $s_1$  even.

You are allowed to assume the following lemmas:

- **Inversion for nil:** For all  $s_1$  and  $s_2$ , if  $\text{cut}(s_1, s_2, \text{nil})$ , then  $s_1 = s_2$  and  $s_1$  stack.
- **Inversion for cons:** For all  $s_1$ ,  $s_2$ , and  $s_3$ , if  $\text{cut}(s_1, s_2, \text{cons}(c, s_3))$ , then there exists a  $s'_1$  such that  $s_1 = \text{cons}(c, s'_1)$ ,  $c$  card, and  $\text{cut}(s'_1, s_2, s_3)$ .