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.

   Solution 1.1 Dolores' actions were permitted, but Toby's were not – you cannot use notes from a conversation (such as the IMs) while writing a solution.

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.

   Solution 1.1 This is fine; the notes were discarded and a period of time passed before the problem set was turned in.

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.

   Solution 1.1 Ian and Jeremy perhaps obeyed the letter of the policy, but they did not obey it in spirit. If you're "running" to type up the answer, you are less likely to really understand the work that you're submitting. Letting time pass is intended to ensure that you are actually thinking about the problem rather than copying from memory; we would certainly frown on this behavior, even if it is not clearly a violation of the policy.

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.

   Solution 1.1 Nitin's actions were not permitted, you cannot take notes away. Margaret is fine.

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).

$$\frac{}{\heartsuit\ \mathsf{card}}\ (1) \qquad \frac{}{\spadesuit\ \mathsf{card}}\ (2) \qquad \frac{}{\clubsuit\ \mathsf{card}}\ (3) \qquad \frac{}{\diamondsuit\ \mathsf{card}}\ (4)$$

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

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. $\mathcal{P}(\mathsf{nil})$
2. $\mathcal{P}(\mathsf{cons}(c, s))$ 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 $\mathsf{unshuffle}(s_1, s_2, s_3)$ 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.

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

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

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

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

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

Solution 2.2 The two derivations differ based on whether the first spade ends up in the first or second pile. One derivation is:

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

The other derivation is:

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

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 $\mathsf{unshuffle}(s_1, s_2, s_3)$.

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.

$\boxed{\textbf{Solution 2.3}}$ The induction principle tells us that, in order to show that $\mathcal{P}(s)$, it suffices to show

- $\mathcal{P}(\mathsf{nil})$
- $\mathcal{P}(\mathsf{cons}(c, s))$ assuming $c\,\mathsf{card}$ and $\mathcal{P}(s)$

We proceed by rule induction on $s\,\mathsf{stack}$, using $\mathcal{P}(s)$ = "There exist s_2 and s_3 such that $\mathsf{unshuffle}(s, s_2, s_3)$."

- **Show** $\mathcal{P}(\mathsf{nil})$:

  To show: There exist s_2 and s_3 such that $\mathsf{unshuffle}(\mathsf{nil}, s_2, s_3)$
  Take s_2 and s_3 to both be nil
  To show: $\mathsf{unshuffle}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})$
  1) $\mathsf{unshuffle}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})$ — *By rule (7)*

- **Assuming** $\mathcal{P}(s)$ **and** $c\,\mathsf{card}$**, show** $\mathcal{P}(\mathsf{cons}(c, s))$: Assume that there exist s_2' and s_3' such that $\mathsf{unshuffle}(s, s_2', s_3')$, and assume $c\,\mathsf{card}$.

  To show: Exist s_2 and s_3 such that $\mathsf{unshuffle}(\mathsf{cons}(c, s), s_2, s_3)$
  Take s_2 to be $\mathsf{cons}(c, s_2')$ *and take s_3 to be s_3'*
  To show: $\mathsf{unshuffle}(\mathsf{cons}(c, s), \mathsf{cons}(c, s_2), s_3)$
  1) $\mathsf{unshuffle}(s, s_2', s_3')$ — *By induction hypothesis*
  2) $c\,\mathsf{card}$ — *By induction principle*
  3) $\mathsf{unshuffle}(\mathsf{cons}(c, s), \mathsf{cons}(c, s_2'), s_3')$ — *By rule (9) on (1) and (2)*

Induction proves that if $s\,\mathsf{stack}$, there exist s_2 and s_3 such that $\mathsf{unshuffle}(s, s_2, s_3)$, which is what we wanted to prove.

Note: in the inductive step, we could have taken s_2 to be s_2' and s_3 to be $\mathsf{cons}(c, s_3')$, in which case we would have needed to derive the conclusion using rule (8) instead of rule (9).

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 $\diamondsuit$ 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:

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

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 stack 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.

$\boxed{\textbf{Solution 2.4}}$ There are a few ways of doing this. One is:

$$\frac{}{\mathsf{separate}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})}$$

$$\frac{\mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(\heartsuit, s_1), \mathsf{cons}(\heartsuit, s_2), s_3)} \qquad \frac{\mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(\diamondsuit, s_1), \mathsf{cons}(\diamondsuit, s_2), s_3)}$$

$$\frac{\mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(\clubsuit, s_1), s_2, \mathsf{cons}(\clubsuit, s_3))} \qquad \frac{\mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(\spadesuit, s_1), s_2, \mathsf{cons}(\spadesuit, s_3))}$$

Another equally acceptable strategy is to create an iterated inductive definition, starting with two auxiliary judgments c black and c red.

$$\frac{}{\spadesuit\ \mathsf{black}} \qquad \frac{}{\clubsuit\ \mathsf{black}} \qquad \frac{}{\heartsuit\ \mathsf{red}} \qquad \frac{}{\diamondsuit\ \mathsf{red}} \qquad \frac{}{\mathsf{separate}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})}$$

$$\frac{c\ \mathsf{red} \quad \mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(c, s_1), \mathsf{cons}(c, s_2), s_3)} \qquad \frac{c\ \mathsf{black} \quad \mathsf{separate}(s_1, s_2, s_3)}{\mathsf{separate}(\mathsf{cons}(c, s_1), s_2, \mathsf{cons}(c, s_3))}$$

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?

$\boxed{\textbf{Solution 2.5}}$ This mode means that, for all s_1 stack, there exists a **unique** s_2 and s_3 such that $\mathsf{separate}(s_1, s_2, s_3)$. Another way of saying this is that this mode makes two statements:

- **Existence**: For all s_1 stack, there exists s_2 and s_3 such that $\mathsf{separate}(s_1, s_2, s_3)$
- **Uniqueness**: For all s_1 stack and for all derivations of $\mathsf{separate}(s_1, s_2, s_3)$, and for all derivations of $\mathsf{separate}(s_1, s_2', s_3')$, $s_2 = s_2'$ and $s_3 = s_3'$.

The relation unshuffle does not have this mode because, for most stacks s_1 (indeed, for any stack that is not nil), we can prove $\mathsf{unshuffle}(s_1, s_2, s_3)$ for **different** s_2 and s_3.

As a trivial example, both of these are provable:

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

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

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.

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

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.

Solution 3.1

In order to prove that $\mathcal{P}_{\text{even}}(s)$ whenever s even and $\mathcal{P}_{\text{odd}}(s)$ whenever s odd, it is enough to show the following:

- $\mathcal{P}_{\text{even}}(\mathsf{nil})$
- If c card and $\mathcal{P}_{\text{odd}}(s)$, then $\mathcal{P}_{\text{even}}(\mathsf{cons}(c, s))$
- If c card and $\mathcal{P}_{\text{even}}(s)$, then $\mathcal{P}_{\text{odd}}(\mathsf{cons}(c, s))$

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.

Solution 3.2

The tricky part here is to figure out exactly what to say for $\mathcal{P}_{\text{odd}}$. Here are induction hypotheses that works:

- $\mathcal{P}_{\text{odd}}(s)$ is "s stack"
- $\mathcal{P}_{\text{even}}(s)$ is "s stack"

We perform induction to show that whenever s even, s stack, and whenever s odd, s stack.

- **Show $\mathcal{P}_{\text{even}}(\mathsf{nil})$:** We have to show nil stack; this is immediate from rule (5).
- **Assuming c card and $\mathcal{P}_{\text{odd}}(s)$, show $\mathcal{P}_{\text{even}}(\mathsf{cons}(c, s))$:** Assume c card and s stack for an arbitrary c and s. We have to show $\mathsf{cons}(c, s)$ stack; this can be derived using rule (6).
- **Assuming c card and $\mathcal{P}_{\text{even}}(s)$, show $\mathcal{P}_{\text{odd}}(\mathsf{cons}(c, s))$:** Assume c card and s stack for an arbitrary c and s. We have to show $\mathsf{cons}(c, s)$ stack; this can be derived using rule (6).

Induction gives us that for all s, whenever s even, s stack, and whenever s odd, s stack. Because we wanted to show that for all s whenever s even then s stack, we are done.

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

For all S, if

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

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 $S(\mathsf{cons}(c', s))$."

Solution 3.3

We want to show that for some S, if $S(\mathsf{nil})$ and if for all c_1, c_2, and s, $S(s)$ implies $S(\mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$, then for all s, if s even then $S(s)$.

We start, therefore, by assuming the existence of some arbitrary $S(x)$, and then assuming that $S(\mathsf{nil})$ (we will refer to this assumption as **Base**, for "base case"), and furthermore assuming that for all c_1, c_2 and s, if c_1 card and c_2 card and $S(s)$, then $S(\mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$ (we will refer to this assumption as **Ind**, for "inductive case").

Now, we need to show that if s even then $S(s)$. We will prove this by using rule induction. The induction hypothesis we use is the following:

- $\mathcal{P}_{\mathsf{even}}(s)$ is simply "$S(s)$"
- $\mathcal{P}_{\mathsf{odd}}(s)$ is "For all c', if c' card, then $S(\mathsf{cons}(c', s))$"

Now, we proceed with induction:

- **Show** $\mathcal{P}_{\mathsf{even}}(\mathsf{nil})$: We need to show $S(\mathsf{nil})$, but this is true immediately from assumption **Base**.
- **Assuming** c card **and** $\mathcal{P}_{\mathsf{odd}}(s)$**, show** $\mathcal{P}_{\mathsf{even}}(\mathsf{cons}(c, s))$:

  To show: $S(\mathsf{cons}(c, s))$

1) c card	*By induction principle*
2) For all c', if c' card then $S(\mathsf{cons}(c', s))$	*By induction hypothesis*
3) If c card, then $S(\mathsf{cons}(c, s))$	*By (2), taking c' to be c*
4) $S(\mathsf{cons}(c, s))$	*By (3) on (1)*

- **Assuming** c card **and** $\mathcal{P}_{\mathsf{even}}(s)$**, show** $\mathcal{P}_{\mathsf{odd}}(\mathsf{cons}(c, s))$:

  To show: For all c', if c' card then $S(\mathsf{cons}(c', \mathsf{cons}(c, s)))$

1) c card	*By induction principle*
2) $S(s)$	*By induction hypothesis*
3) c' card	*By assumption*
4) If c' card and c card and $S(s)$, then $S(\mathsf{cons}(c', \mathsf{cons}(c, s)))$	*By **Ind**, taking c_1 to be c' c_2 to be c, and s to be s*
5) $S(\mathsf{cons}(c', \mathsf{cons}(c, s)))$	*By (4) on (3) and (1) and (2)*

By induction, we have shown that if s even then $S(s)$, and if s odd then for all c', if c' card then $S(\mathsf{cons}(c', s))$. Because we needed to show that if s even then $S(s)$, we are done.

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

To show that $\mathcal{S}(s)$ whenever s even, it is enough to show

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

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.

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

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 $\mathsf{cut}(s_1, s_2, s_3)$, then s_1 even.

You are allowed to assume the following lemmas:

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

Solution 3.4 We perform rule induction over s_3 even, and the induction hypothesis $\mathcal{S}(s)$ is "For all s_1 and s_2, if s_2 even and $\mathsf{cut}(s_1, s_2, s)$, then s_1 even."

1. **Show** $\mathcal{S}(\mathsf{nil})$:

   To show: If s_2 even *and* $\mathsf{cut}(s_1, s_2, \mathsf{nil})$, *then* s_1 even

1) s_2 even	*By assumption*
2) $\mathsf{cut}(s_1, s_2, \mathsf{nil})$	*By assumption*
To show: s_1 even	
3) s_1 and s_2 are identicial and s_1 stack	*By inversion for* nil *on (2)*
To show: s_2 even *(because s_1 and s_2 are identical)*	
4) s_2 even	*By (1)*

2. **Given** c_1 card, c_2 card, **and** $\mathcal{S}(s)$, **show** $\mathcal{S}(\mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$:

1) c_1 card	*By induction principle*
2) c_2 card	*By induction principle*
3) For all s_1^* and s_2^*, if s_2^* even and $\mathsf{cut}(s_1^*, s_2^*, s)$, then s_1^* even	*By induction hypothesis*
To show: For all s_1 and s_2, if s_2 even *and* $\mathsf{cut}(s_1, s_2, \mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$, *then* s_1 even	
4) s_2 even	*By assumption*
5) $\mathsf{cut}(s_1, s_2, \mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$	*By assumption*
To show: s_1 even	
6) $s_1 = \mathsf{cons}(c_1, s_1')$, c_1 card, and $\mathsf{cut}(s_1', s_2, \mathsf{cons}(c_2, s))$	*By inversion for* cons *on (5)*
7) $s_1' = \mathsf{cons}(c_2, s_1'')$, c_2 card, and $\mathsf{cut}(s_1'', s_2, s)$	*By inversion for* cons *on (6)*
To show: $\mathsf{cons}(c_1, \mathsf{cons}(c_2, s_1''))$ even	
8) If s_2 even and $\mathsf{cut}(s_1'', s_2, s)$, then s_1'' even	*By (3), taking $s_1^* = s_1''$ and $s_2^* = s_2$*
9) s_1'' even	*By (8) on (4) and (7)*
10) $\mathsf{cons}(c_2, s)$ odd	*By rule (12) on (2) and (9)*
11) $\mathsf{cons}(c_1, \mathsf{cons}(c_2, s))$ even	*By rule (11) on (1) and (10)*

From induction, we know that for all s_3, whenever s_3 even, then for all s_1 and s_2, if s_2 even and $\mathsf{cut}(s_1, s_2, s_3)$ then s_1 even. We were asked to prove that for all s_1, s_2, s_3, if s_2 even and s_3 even and $\mathsf{cut}(s_1, s_2, s_3)$, then s_1 even. We could just say "rearringing the premises, we are done," because while these statements are not identitical, they are close to identical. If we want to be totally formal, we can actually prove one using the other:

To show: For all s_1^*, s_2^*, s_3^*, *if* s_2^* even *and* s_3^* even *and* $\mathsf{cut}(s_1^*, s_2^*, s_3^*)$, *then* s_1^* even	
1) s_2^* even	*by assumption*
2) s_3^* even	*by assumption*
3) $\mathsf{cut}(s_1^*, s_2^*, s_3^*)$	*by assumption*
To show: s_1^* even	
4) If s_3^* even, then for all s_1 and s_2, if s_2 even and $\mathsf{cut}(s_1^*, s_2, s_3)$, then s_1^* even.	*by proof above, taking* s_1 *to be* s_1^*
5) For all s_1 and s_2, if s_2 even and $\mathsf{cut}(s_1^*, s_2, s_3)$, then s_1^* even	*by (4) on (3)*
6) If s_2^* even and $\mathsf{cut}(s_1^*, s_2^*, s_3^*)$, then s_1^* even	*By (5), taking* s_2 *to be* s_2^* *and* s_3 *to be* s_3^*
7) s_1^* even	*By (6) on (1) and (3)*