CS101 Homework Assignment 11

Tadius Frank

December 4, 2021

1. **Academic Honesty Declaration**

   In the process of finishing this homework:

   (a) I had conversations about the contents and solutions of this assignment with the following people:

   (b) I consulted the following resources, such as books, articles, webpages:

   - https://wch.github.io/latexsheet/
   - https://en.wikibooks.org/wiki/LaTeX/Basics

   (c) I did not look at the answers of any other students.

   (d) I did not provide my answers to other students.

2. **Writing Component**

(a) **Semi-Decidability**

One of the following sets is semi-decidable, and the other is not. Which is which? Give proofs for both.
$L_1 = \{M \mid L(M)$ contains at least 481 elements$\}$
$L_2 = \{M \mid L(M)$ contains at most 480 elements$\}$

Solution:

L_1 is semi-decidable because if simulate a Turing Machine M on inputs from L_1 and it would halt and accept on the 481th element in the set that is in $L(M)$. We could have more inputs in L_1 (since it contains at least 481 elements meaning that they could be more than 481 inputs to that can be accepted by our machine) that are in $L(M)$ but we don't need to worry about them as long as we have have 481 elements already. So, we could run forever since non of these inputs will accept, and halt and return at some point. For L_2, let us assume that L_2 is decidable and go forth with a proof by contradiction.

i. Then there exists a function G that returns true iff L(M) contains at 480, so there exists a Turing Machine, V, that computes G.

ii. We can construct a Turing Machine, M, such that it takes as an input the encoding of a machine p and an input w, which discards its own input and simulates p on w. We'll call this machine $M_{<p,w>}$.

iii. If p halts on w, and $L(M_{<p,w>})$ is everything and if p does not halt on w, $L(M_{<p,w>})$ = empty.

iv. If $M_{<p,w>}$ accepts everything it also halts and p does to halt on w, $V(M_{<p,w>})$ rejects it because it would be infinite.

v. If $M_{<p,w>}$ is empty which means it accepts nothing and so it does not halt, $V(M_{<p,w>})$ accepts because this input is finite and it halts on it.

vi. There's a contradiction! We technically solved the halting problem because we have constructed a machine that accepts and halts on an input which does not halt.

vii. However, we know the halting problem is undecidable, so our assumption that L_2 is decidable must be incorrect.

So, we can conclude that L_2 is undecidable.

(b) **Undecidability**

In this problem, we would like you to show the set is not decidable.
$L_{Regular}\{< M > \mid M$ is a TM *and* $L(M)$ is regular $\}$
Do not use Rice's theorem.

Solution:

i. Assume there exists a Turing Machine $M_{<p,w>}$ such that it takes as an input the encoding of a machine p and an input w, which discards its own input and simulates p on w.

ii. If p halts on w, when $L(M_w >)$ is everything and M_w accepts it.

iii. Otherwise, it loops forever, in this case we introduce an arbitrary input x then we simulate our Machine $M_{<p,w>}$ on x to check if it's in the non-regular language $0^n1^n0^n$.
- If $M_{<p,w>}$ accepts and halts x, then x is not in in the non-regular language $0^n1^n0^n$.
- If $M_{<p,w>}$ does not halt on x, then x is in the non-regular language $0^n1^n0^n$, so $L(M_w)$ must be irregular.

iv. Now assume, there's a Turing Machine V which takes the Turing Machine $M_{<p,w>}$ as input.

v. If $V(M_{<p,w>})$ says yes, then p halts on w and $L(M_{<p,w>})$ is regular.

vi. If $V(M_{<p,w>})$ says no, then p does not halt on w and $L(M_{<p,w>})$ is not regular.

vii. There's a contradiction! We technically solved the halting problem because we have constructed a machine that accepts and halts on an input which does not halt.

viii. However, we know the halting problem is undecidable, so our assumption that $L_{Regular}$ is decidable must be incorrect.

(c) **Primitive recursive functions**

(a) Please write the primitive recursive definition of the function isZero(x) where isZero(x) = 1 if x is 0, and = 0 otherwise.

Solution:
isZero(0) = 1
isZero(x + 1) = 0

(b) Please write the primitive recursive definition of the function monus(m,n) where monus (m,n) = m - n if m $\geq$ n and 0 otherwise.

Solution:
monus(m,0) = m
monus(m,n+1) = pred(monus(m,n))

(c) Please write the primitive recursive definition of the function lessOrEq where lessOrEq (m,n) = 1 if m $\leq$ n and 0 otherwise.
Solution:

lessOrEq(m,n) = isZero(monus(m,n))

(d) **Lambda Reduction with Sugar**

Simplify the de-sugared lambda expression using reduction. Show all of your reduction steps. Write one or two sentences explaining why the simplified expression is the answer you expected.

Solution:

I utilized beta reduction to process the de-sugared lambda expression. This is the process of calling the lambda expression with input, and getting the output.

3. **Type checkers**
   Begin by writing the type-checking rule for let expressions. This should be similar to the rules for other constructs expressed in the last few type-checking slides of Lecture 21. Pay special attention to the type-checking rules for function definition and application. Include this answer with your written problems for this week.

   Solution:

$$\frac{E \cup \{vble : T\} \vdash body : B, E \vdash term : T}{E \vdash let\ (vble : T) = term\ in\ body : B}$$

Extra