Models of Software Systems	Name: Danila Danko
Assignment 1	

1 Lambda Language

Consider a language with alphabet $\{\lambda, \bullet, (,), x, y, z\}$ and syntax

$$\begin{aligned}
expression \quad &= variable\ name \mid expression, expression \\
&\mid \text{``}\lambda\text{''}, variable\ name, \text{``}\bullet\text{''}, expression \\
&\mid \text{``(''}, expression, \text{``)''}; \\
variable\ name &= \text{``}x\text{''}\mid\text{``}y\text{''}\mid\text{``}z\text{''};
\end{aligned}$$

Are the following wffs of the language? For those that are not briefly explain why.

(a) $\lambda\, x \bullet yz$ Yes

(b) $(x)(y)$ Yes

(c) $\lambda \bullet x\, \lambda \bullet y$ No - There must be a *variable name* after a λ

(d) $(\lambda\, x \bullet x(yz))$ Yes

(e) $\lambda\, x \bullet \lambda\, y \bullet xyz$ Yes

2 Stars, Derivation

Using the *Stars* formal system that was discussed during the tutorial on week 2 formally shows that *◇**◦***** ⊢ *◇****◦*******

2.1 Proof

1. *◇**◦*****   axiom A (because it's a wff of *Stars*)

2. *◇***◦******   R, 1

3. *◇****◦*******   R, 2

3 Prolog

(a) Implement the problem "Tower of Hanoi" in Prolog. For information about the problem, visit the link below: $https://en.wikipedia.org/wiki/Tower\_of\_Hanoi$

(b) Implement a program in Prolog that takes two lists of integers and returns a new list containing only the elements that appear in both input lists.

(c) Implement a Prolog program that checks if a given list is a palindrome.