Homework: Syntax and Semantics with Functions
Optional Assignment A: Can be used to replace your lowest scoring homework assignment

Due-date: April 10 at 11:59pm
Submit online on Blackboard LS

Homework must be individual's original work. Collaborations and of any form with any students or other faculty members are not allowed. If you have any questions and/or concerns, post them on Piazza and/or ask our TA or me.

Learning Outcomes

- Knowledge and application of Functional Programming
- Ability to understand syntax specification
- Ability to design software following requirement specifications (operational semantics)

Questions

Consider the grammar G of a language $\mathcal{L}$, where $G = (\Sigma, V, S, P)$ such that

- Σ is a set of terminals: anything that does not appear on the left-side of the product rules P presented below
- V is the set of non-terminals appearing in the left-side of the production rules P presented below

```
Program      -> Expr
Expr         -> Number | Variable | OpExpr | FExpr | ApplyF
                |DRef | WRef | Ref
OpExpr       -> ArithExpr | CondExpr | VarExpr
ArithExpr    -> (Op Expr Expr)
Op           -> + | -  | * | /
CondExpr     -> (CCond Expr Expr)
CCond        -> BCond | (or CCond CCond) | (and CCond CCond) | (not CCond)
BCond        -> (gt Expr Expr) | (lt Expr Expr) | (eq Expr Expr)
VarExpr      -> (var VarAssign Expr)
VarAssign    -> (VarAssignSeq)
VarAssignSeq -> (Variable Expr) | (Variable Expr) VarAssignSeq
Variable     -> symbol
FExpr           -> (fun FAssign Expr)
FAssign         -> ((FName FormalParams) Expr)
FormalParams    -> () | (FormalParamList)
FormalParamList -> Variable | Variable FormalParamList
ApplyF          -> (apply (FName Args))
Args            -> () | (ArgsList)
```

```
ArgList          -> Expr | Expr ArgList
FName            -> symbol
DRef       -> (deref Expr)
WRef       -> (wref Expr Expr)
Ref        -> (ref Expr) | (free Expr)
```

- $S =$ `Program`

We will define the following reference-type concept for the valid expressions of program written using the above grammar.

- The reference-type of a variable is true if and only if it is assigned to some expression, whose reference-type is true
- The reference-type of an arithmetic expression is true if and only if the reference-type of at least one of the operands is true
- The reference-type of a conditional expression is true if and only if the reference type of at least one of then-expression or else-expression is true
- The reference-type of an application expression is true if and only if the reference-type of the definition of the function being applied is true
- The reference-type of ref-expression and free-expressions is true

1. Write a function `reftype` that takes as input a valid program and returns true if and only if the reference-type of the program is true. You should write your function in a file named `<netid>.rkt`. At the top of this file, you must include

   ```
   #lang racket
   (require "program.rkt")
   (provide (all-defined-out))
   ```

2. Does the reference-type depend on the static and dynamic scoping semantics of function? Justify your response.

   Write your answer as Racket comment in the file `<netid>.rkt`. You can use multi-line comments enclosed in $\#|$ and $|\#$.

You are required to submit the `<netid>.rkt` file only on the blackboard. If you netid is asterix, then the name of the file should be asterix.rkt. You must follow the instructions for submission. For instance, do not submit the file with any other extension, do not submit any other file, do not submit any zip files, remove/comment any inclusion of test code including trace directives that get automatically executed when your code is tested for evaluation, remove incomplete code.

In terms of the racket language features you can use, the same rules continue to apply for this assignment as well: list, numbers, boolean and typical operations over them, if-then-else and cond.

If you do not follow the requirements or submission guidelines of the assignment, then your submission may not graded. If you have doubts about the instruction, please post/ask.