HW#2: Syntactic sugar, desugaring, and an infix parser (5 points)

Goal: Through this HW,
(1) you can clearly understand concepts of syntactic sugar and desugaring in programming languages. https://en.wikipedia.org/wiki/Syntactic_sugar
(2) implement a infix parser to check if you have basic understanding of a parser

For this homework, you are not allowed to discuss with anybody except for JC and TAs. You can send JC and TAs emails and ask in our chatroom and/or use a FAQ doc. Do not simply ask if your answers are correct or not!! Bring detailed questions and your concerns.

Tasks

Task 1: Read this chapter and answer the following questions: http://cs.brown.edu/courses/cs173/2012/book/first-desugar.html

(Since the second edition use #lang plai-typed, the code in the link uses a different grammar. For example, when using #lang plai, the ArithC is defined like this.

```
; Data type for ArithC
(define-type ArithC
  [numC (n number?)]
  [plusC (l ArithC?) (r ArithC?)]
  [multC (l ArithC?) (r ArithC?)])
```

You can try to run the code in the link after converting or use #lang play-typed for your practice.)

* **Copy this doc into your google drive for this class** and **name it "HW2-your-student-id"**. Do not download it as a docx file but **edit the copied google doc in your browser**. Without an editing history of your google doc, it will be considered as cheating or you will lose your mark.

P1. Provide two examples of syntactic sugar of your favorite languages (C, Java, python, JavaScript, etc.) and explain your thoughts why they are synthetic sugar in detail (based on the chapter 4). (1 points) * You do not need to find a real syntactic sugar but just guessing them is also fine. If you can provide reasonable/logical explanations about them, you can get points regardless of whether they are really implemented as syntactic sugar or not.)

Answer:

P2. For your favorite languages, can you create new syntactic sugar that can make you and other developers happy? Please, design a new syntactic sugar in BNF (a partial BNF is fine) and provide an example code for both sugaring and desugaring) (2 points)

Answer:

Task 2: Write an infix parser and an interpreter for the AE language

* After implementing the parser and interpreter, upload "HW2-InfixParser-YourStudentId.rkt" into your shared google drive for the PLT class.
An example concrete code in the infix style: {{3 + 4} + {5 - 2}}
The example code in abstract syntax: (add (add (num 3) (num 4)) (sub (num 5) (num 2)))
- The function name for a parser: infix-parse
- The function name for an interpreter: interp
* For function definitions, please add [contract], [purpose], [tests] (define at least five test cases that test unique cases) as comments. e.g., (test (interp (infix-parse ...)
*** For the implementation of an infix parser, use "match" ~~the type deconstruction~~!** (Additional requirement)

Due Date

1. Upload your code file
    a. 22:00, Oct ~~5~~ 7 (Tue) 2021.

Evaluation (Full mark: 5 points)

- Late submission
    - No score.
- P1
    - Wrong answer: -1
    - Partially correct: -0.5
- P2
    - Wrong: -2
    - Partially correct: -1
- Task2
    - Wrong implementation and cannot compile: -2
    - Failed test cases: -0.5 per each
    - Works but weird implementation (not fully understand a concept of a parser and an interpreter: -1

HW2 FAQ

iIf you have any questions, put your questions in HW2 FAQ by creating a comment and tag JC(+jcnam@handong.edu) or TAs.
https://docs.google.com/document/d/1fAdLdFc9-Nc9odOhKgDgDMEROVVL0vos_1r4Vi8vT3c/edit#