Name:

CS 443 — Follow Set Generator

Mattox Beckman

Objectives

- Understand how to determine the follow set of a grammar. simultaneously.

Given Code

Here is a part of the code that generated the first sets.

```
(ns grammars.core
  (:require [clojure.core.typed :as t])
  (:gen-class))

(def ex1 {'S [['x 'S 'x]
              [:eps]]})

(def fs1 {'S {:terms #{'x}
              :nonterms #{'E}
              :eps false}
          'E {:terms #{'y}
              :nonterms #{}
              :eps true}})

(defn join-fs [fs1 fs2]
  {:terms (into (:terms fs1) (:terms fs2))
   :nonterms (into (:nonterms fs1) (:nonterms fs2))
   :eps (or (:eps fs1) (:eps fs2))})

(defn first-set-of [rule fs-map]
  (loop [r rule
         acc  {:terms []
               :nonterms []
               :eps false}]
    (if (empty? r) acc
        (let [s (first r)
              fs (fs-map s)]
          (cond (nil? s)  acc
                (nil? fs) (update-in acc [:terms] #(conj % s))
                (:eps fs) (recur (rest rule)
                                 (join-fs acc fs))
                :elsa     (join-fs acc fs))))))
```

Your Work

- Write a function `gen-follow-set` that is analogous to the `gen-first-set` done in class. It will need as inputs a grammar and its first sets.