Parser Language

Give the following language syntax:

G’: L -> E L'

R -> E L' | ϵ

E -> ( E' | V | T

Z -> C ) | F )

C -> if E E C'

X -> ϵ | E

F -> + L | - L | \* L | print L

V -> a | b | c | d

T -> 0 | 1 | 2 | 3

We are able to construct the follow parse table by using first and follow sets:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | 0 | 1 | 2 | 3 | + | − | \* | print | ( | ) | $ | if |
| L | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ |  |  |  |  | EL’ |  |  |  |
| R | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ | EL’ |  |  |  |  | EL’ | ε | ε |  |
| E | V | V | V | V | T | T | T | T |  |  |  |  | (E’ |  |  |  |
| Z |  |  |  |  |  |  |  |  | F) | F) | F) | F) |  |  |  | C) |
| C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ifEEC’ |
| X | E | E | E | E | E | E | E | E |  |  |  |  | E | ϵ |  |  |
| F |  |  |  |  |  |  |  |  | + L | − L | \*L | printL |  |  |  |  |
| V | a | b | c | d |  |  |  |  |  |  |  |  |  |  |  |  |
| T |  |  |  |  | 0 | 1 | 2 | 3 |  |  |  |  |  |  |  |  |

The way this parse table works as follows:

* The first column represents available variables
* The first row represents available terminals
* Given a variable, substitute it by the element attained by the desired terminal

e.g. Given variable “L” and desired terminal “3”

(map L and 3 on parse table for substitution)

L = EL’

* Blank spots on parse table represent terminals which the variable cannot reach

e.g. Variable “Z” does not have a substitute for terminal “a”

* Therefore, given a string of terminals, we can determine whether the string follows the syntax for our given language by continuously substituting the variables until the string is achieved
* If we are to reach a “blank spot” by attempting to substitute a terminal into a variable, then the language does not accept that particular string