عرفان افشار ۹۶۳۱۰۰۷

گزارش کد تبدیل زبان نوع دوم به فرم نرمال چامسکی

بخش ۱ : کد

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws FileNotFoundException {
        File file_g = new File("grammar.txt");
        File file_t = new File("terminals.txt");
        File file_v = new File("variables.txt");

        Scanner scanner_g = new Scanner(file_g);
        Scanner scanner_t = new Scanner(file_t);
        Scanner scanner_v = new Scanner(file_v);

        String[] variables = scanner_v.nextLine().split(",");
        String[] terminals = scanner_t.nextLine().split(",");
        String[] newVariables = new String[26 - variables.length];

        char character;
        boolean found;
        int count = 0;
        for (int i = 65; i < 91; i++) {
            found = false;
            character = (char) i;
            for (int j = 0; j < variables.length; j++) {
                if (variables[j].equals(Character.toString(character))) {
                    found = true;
                    break;
                }
            }
            if (!found) {
                newVariables[count] = Character.toString(character);
                count++;
            }
        }

        int nOfL = 0;
        while (scanner_g.hasNextLine()) {
            scanner_g.nextLine();
            nOfL++;
        }

        scanner_g.close();
        scanner_g = new Scanner(file_g);
        String[] grams = new String[nOfL];
        String[] tmp;
        String[] sGram = new String[nOfL];
        String[] eGram = new String[nOfL];
        int nOfOr = 0;
        int[] orInL = new int[nOfL];

        for (int i = 0; i < nOfL; i++) {
            grams[i] = scanner_g.nextLine();
            for (int j = 0; j < grams[i].length(); j++) {
                if (grams[i].charAt(j) == '|') {
                    orInL[i]++;
```

```
                    nOfOr++;
                }
            }

            tmp = grams[i].split("->");
            sGram[i] = tmp[0];
            eGram[i] = tmp[1];
        }


        String[] iGram = new String[nOfL + nOfOr];
        String[] oGram = new String[nOfL + nOfOr];
        for (int i = 0, j = 0; (i < nOfL) && (j < nOfL + nOfOr); i++, j++) {
            if (orInL[i] == 0) {
                iGram[j] = sGram[i];
                oGram[j] = eGram[i].substring(0, eGram[i].length() - 1);
            } else {
                String[] outs;
                outs = eGram[i].substring(0, eGram[i].length() - 1).split("\\|");
                for (int k = 0; k < orInL[i] + 1; k++, j++) {
                    iGram[j] = sGram[i];
                    oGram[j] = outs[k];
                }
                j--;
            }
        }

        String[] iChomsky = new String[30];
        String[] oChomsky = new String[30];
        calcChomsky(iGram, oGram, iChomsky, oChomsky, terminals, newVariables);
    }

    private static void calcChomsky(String[] iGram, String[] oGram, String[] iChomsky,
                                    String[] oChomsky, String[] terminals, String[]
newVariables) {

        String[] iChomsky1 = new String[30];
        String[] oChomsky1 = new String[30];
        int k = 0;

        for (int i = 0; i < oGram.length; i++) {
            if (oGram[i].length() == 1) {
                if (Character.isUpperCase(oGram[i].charAt(0))) {
                    for (int j = 0; j < iGram.length; j++) {
                        if (iGram[j].equals(oGram[i].substring(0, 1))) {
                            iChomsky1[k] = iGram[i];
                            oChomsky1[k] = oGram[j];
                            k++;
                        }
                    }
                } else {
                    iChomsky1[k] = iGram[i];
                    oChomsky1[k] = oGram[i];
                    k++;
                }
            } else {
                iChomsky1[k] = iGram[i];
                oChomsky1[k] = oGram[i];
                k++;
            }
        }

        String[] iChomsky2 = new String[30];
```

```
        String[] oChomsky2 = new String[30];
        k = 0;

        for (int i = 0; i < oChomsky1.length; i++) {
            if (oChomsky1[i] != null) {
                if (oChomsky1[i].length() == 1) {
                    if (Character.isLowerCase(oChomsky1[i].charAt(0))) {
                        iChomsky2[k] = iChomsky1[i];
                        oChomsky2[k] = oChomsky1[i];
                        k++;
                    } else {
                        System.out.println("error");
                    }
                } else {
                    iChomsky2[k] = iChomsky1[i];
                    oChomsky2[k] = oChomsky1[i];
                    for (int j = 0; j < terminals.length; j++) {
                        if (oChomsky2[k].contains(terminals[j])) {
                            oChomsky2[k] = oChomsky2[k].replace(terminals[j],
newVariables[j]);
                        }
                    }
                    k++;
                }
            }
        }

        k = 0;
        int t = terminals.length;

        for (int i = 0; i < oChomsky2.length; i++) {
            if (oChomsky2[i] != null) {
                if (oChomsky2[i].length() <= 2) {
                    iChomsky[k] = iChomsky2[i];
                    oChomsky[k] = oChomsky2[i];
                    k++;
                } else {
                    for (int j = 0; j < oChomsky2[i].length() - 2; j++) {
                        if (j == 0) {
                            iChomsky[k] = iChomsky2[i];
                            oChomsky[k] = oChomsky2[i].substring(j, j +
1).concat(newVariables[t]);
                            t++;
                            k++;
                        } else {
                            iChomsky[k] = newVariables[t - 1];
                            oChomsky[k] = oChomsky2[i].substring(j, j +
1).concat(newVariables[t]);
                            t++;
                            k++;
                        }
                    }
                    iChomsky[k] = newVariables[t - 1];
                    oChomsky[k] = oChomsky2[i].substring(oChomsky2[i].length() - 2,
oChomsky2[i].length());
                    k++;
                }

            }
        }

        for (int j = 0; j < terminals.length; j++) {
            iChomsky[k] = newVariables[j];
```

```
            oChomsky[k] = terminals[j];
            k++;
        }

        for (int i = 0; i < iChomsky.length; i++) {
            if (iChomsky[i] != null) {
                System.out.println(iChomsky[i] + " -> " + oChomsky[i]);
            }
        }
    }
}
```

---

بخش ۲ : بررسی خروجی های تست کیس ها

ورودی ۱ :

S->aSb|Sab|ab;

خروجی ۱ :

S -> AC

C -> SB

S -> SD

D -> AB

S -> AB

A -> a

B -> b

---

ورودی ۲ :

S->aSaaA|A;

A->abA|bb;

خروجی ۲ :

S -> BD

D -> SE

E -> BF

F -> BA

S -> BG

G -> CA

S -> CC

A -> BH

H -> CA

A -> CC

B -> a

C -> b

---

ورودی ۳ :

S->a|aA|B|C;

A->aB|a;

B->Aa;

C->cCD|c;

D->ddd;

خروجی ۳ :

S -> a

S -> EA

S -> AE

S -> GI

I -> CD

S -> c

A -> EB

A -> a

B -> AE

C -> GJ

J -> CD

C -> c

D -> HK

K -> HH

E -> a

F -> b

G -> c

H -> d

---

ورودی ۴ :

S->ABacb|ABa|b;

A->aA|b;

B->bB|b;

خروجی ۴ :

S -> AF

F -> BG

G -> CH

H -> ED

S -> AI

I -> BC

S -> b

A -> CA

A -> b

B -> DB

B -> b

C -> a

D -> b

E -> c

---

ورودی ۵ :

S->TaXU|aXU|TaX|aX;

T->UU|U|abc;

U->bSc;

X->bT|Tc|b|c;

خروجی ۵ :

S -> TD

D -> AE

E -> XU

S -> AF

F -> XU

S -> TG

G -> AX

S -> AX

T -> UU

T -> BH

H -> SC

T -> AI

I -> BC

U -> BJ

J -> SC

X -> BT

X -> TC

X -> b

X -> c

A -> a

B -> b

C -> c

---

ورودی ۶ :

S->AS|ASB|SB;

A->XAS|XS|a;

B->SYS|VV|XAS|XS|a;

X->a;

Y->b;

V->b;

خروجی ۶ : (این گرامر هیچ رشته ای تولید نمی کند چون S قابل حذف کردن نیست)

S -> AS

S -> AE

E -> SB

S -> SB

A -> XF

F -> AS

A -> XS

A -> a

B -> SG

G -> YS

B -> VV

B -> XH

H -> AS

B -> XS

B -> a

X -> a

Y -> b

V -> b

C -> a

D -> b