Eduardo Gabriel Nunes de Farias

Exercício de Compiladores

Gramática:

(1) Calc = Ea '=' {printf("%f", Ea.val);}
(2) Ea = Ta {Ear.vh = Ta.val } Ear {Ea.val = Ear.vs}
(3) Ear = '+' Ta {Ear1.vh = Ear.vh + Ta.val} Ear {Ear.vs=Ear1.vs}
(4) Ear = '-' Ta {Ear1.vh = Ear.vh - Ta.val} Ear {Ear.vs=Ear1.vs}
(5) Ear = $ {Ear.vs = Ear.vh}
(6) Ta = Fa {Tar.vh = Fa.val} Tar {Ta.val = Tar.vs}
(7) Tar = '*' Fa {Tar1.vh = Tar.vh * Fa.val} Tar {Tar.vs=Tar1.vs}
(8) Tar = '/' Fa {Tar1.vh = Tar.vh / Fa.val} Tar {Tar.vs=Tar1.vs}
(9) Tar = $ {Tar.vs=Tar.vh}
(10) Fa = '(' Ea ')' {Fa.val = Ea.val}
(11) Fa = 'cteN' {Fa.val = atof(cteN.lex);}

Analisador preditivo recursivo:

```
void Calc() {
    if (token.categ == atr) {
        token.next();
        printf("%f", Ea());
    } else {
        printf("Erro.");
    }
}

float Ea() {
    return(Ear(Ta()));
}

float Ear(float Earvh) {
    float Earvs;

    if (token.categ == mais) {
        token.next();
        Earvs = Ear(Earvh + Ta());
    } else if (token.categ == menos) {
        token.next();
        Earvs = Ear(Earvh - Ta());
    } else {
        Earvs = Earvh;
    }

    return Earvs;
}

float Ta() {
    return Tar(Fa());;
}
```

```
float Tar(float Tarvh) {
    float Tarvs;

    if (token.categ == mult) {
        token.next();
        Tarvs = Tar(Tarvh * Fa());
    } else if (token.categ == div) {
        token.next();
        Tarvs = Tar(Tarvh / Fa());;
    } else {
        Tarvs = Tarvh;
    }

    return Tarvs;
}

float Fa() {
    if (token.categ == parAbre) {
        token.next();

        if (token.categ == parFecha) {
            token.next();
            return Ea();
        } else {
            printf("Erro.");
        }
    } else if (token.categ == cten) {
        token.next();
        return atof(token.lex);
    } else {
        printf("Erro.");
    }
}
```