

Dada a gramática com as regras de produção:

```
1 Res = Ea '='
2 Ea = Ea '+' Ta
3 Ea = Ea '-' Ta
4 Ea = Ta
5 Ta = Ta '*' Fa
6 Ta = Ta '/' Fa
7 Ta = Fa
8 Fa = '(' Ea ')'
9 Fa = 'cteN'
```

Após remover as recursões à esquerda, temos:

```
1 Res = Ea '='
2 Ea = Ta Ear
3 Ear = '+' Ta Ear
4 Ear = '-' Ta Ear
5 Ear = $
6 Ta = Fa Tar
7 Tar = '*' Fa Tar
8 Tar = '/' Fa Tar
9 Tar = $
10 Fa = '(' Ea ')'
11 Fa = 'cteN'
```

Por fim, vamos aplicar as ações semânticas:

```
1 Res = 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);}
```