



**Universidade Estadual de Santa Cruz – UESC**

**Relatórios de Implementações de Árvore Sintática e Árvore Abstrata  
para GLC**

**Docente César Alberto Bravo Pariente**

**Discente Matheus Miranda Brandão**

**Matrícula 201820065**

**Disciplina Compiladores.**

**Curso Ciência da Computação**

**Semestre 2022.2**

**Ilhéus – BA  
2022**

# Índice

---

<b>GLC LL</b>	<b>3</b>
Produções:	3
<b>Compilando e Executando</b>	<b>4</b>
<b>Testes</b>	<b>5</b>
$m() \{ r(1); \}$	5
$m() \{ h=(x+y); r(0); \}$	5
$m() \{ (1-1); r(1); \}$	6
$m() \{ w(1) \{ (1/x); \}; r(1); \}$	6
$n() \{ (0/y); r(y); \} \quad g() \{ i=y; r(x); \} \quad m() \{ (1-x); r(0); \}$	7
$m() \{ w(x) \{ f(y) \{ k=(1+(1*0)); \}; \}; r(0); \}$	8
<b>Link para download</b>	<b>9</b>
<b>Referências</b>	<b>10</b>

# GLC LL

---

O projeto consiste na implementação em C de um algoritmo que busca simular um Análise Sintática e Árvore de Sintaxe Abstrata que reconhece a linguagem gerada por uma gramática livre de contexto. A execução do código recebe como entrada a palavra e retorna se a palavra foi aceita ou ocorreu um erro, suas produções, a struct referente a Árvore de Análise e a a struct referente a Árvore de Sintaxe Abstrata.

O analisador implementado ignora o token ' '.

## Produções:

$p_1: S \rightarrow M \mid GM \mid NGM$

$p_4: N \rightarrow n() \{ C; r(E); \}$

$p_5: G \rightarrow g() \{ C; r(E); \}$

$p_6: M \rightarrow m() \{ C; r(E); \}$

$p_7: E \rightarrow 0 \mid 1 \mid x \mid y \mid (EXE)$

$p_{12}: X \rightarrow + \mid - \mid * \mid /$

$p_{16}: C \rightarrow h=E \mid i=E \mid j=E \mid k=E \mid z=E \mid (EXE) \mid w(E) \{ C; \} \mid f(E) \{ C; \} \mid o(E;E;E) \{ C; \}$

# Compilando e Executando

---

Para a execução não é necessário o uso de nenhuma dependência, basta compila-lo normalmente.

```
$ gcc proj2_c.c -o proj2_c
```

Ao executar é necessário digitar o nome do arquivo destino contendo as palavras, caso contrário resultará em erro.

Exemplo:

```
$ ./proj2_c examples/inputs.txt
```

Neste projeto pode-se adicionar num .txt todas as palavras separadas por uma quebra de linha. Em caso de erro, pularemos para a próxima palavra.

Exemplo:

```
m(){ r(1); }
m(){ h=(x+y); r(0); }
m(){ (1-1); r(1); }
m(){ w(1) { (1/x); }; r(1); }
n() { (0/y); r(y); } g() { i=y; r(x); } m() { (1-x); r(0); }
m(){w(x){f(y){k=(1+(1*0));};};r(0);}
```

# Testes

---

Caso a entrada dada seja incorreta o programa irá imprimir os tokens até o momento, então avisará sobre o erro, informará qual o código e o token inesperado, então pulará para a próxima palavra. Como outputs temos a palavra analisada, as produções e a árvore sintática (no formato '[index | mapeamento | token]'), foi considerado o pior caso, onde temos uma árvore n-ária de 12. A árvore foi feita seguindo o modelo de árvore compacta.

Para criação de palavras compatíveis com a linguagem gerada pela GLC foi utilizado o website "CFG Developer".

**m(){ r(1); }**

Palavra 1: m(){

Erro 83. Token 'r' inesperado. Prosseguindo para proxima palavra.

Producoes: P1, P6

Arvore sintatica: [0|0| S ], [1|1| M ], [2|13| m ], [3|14| ( ], [4|15| ) ], [5|16| { ], [6|17| C ], [7|18| ; ], [8|19| r ], [9|20| ( ], [10|21| E ], [11|22| ) ], [12|23| ; ], [13|24| } ]

Arvore abstrata: [0 | 0 | m], [1 | 1 | C], [2 | 2 | r], [3 | 5 | E]

**m(){ h=(x+y); r(0); }**

Palavra 2: m(){ h=(x+y); r(0); }

Palavra aceita.

Producoes: P1, P6, P16, P11, P9, P12, P10, P7

Arvore sintatica: [0|0| S ], [1|1| M ], [2|13| m ], [3|14| ( ], [4|15| ) ], [5|16| { ], [6|17| C ], [7|18| ; ], [8|19| r ], [9|20| ( ], [10|21| E ], [11|22| ) ], [12|23| ; ], [13|24| } ], [14|205| h ], [15|206| = ], [16|207| E ], [17|2485| ( ], [18|2486| E ], [19|2487| X ], [20|2488| E ], [21|2489| ) ], [22|29833| x ], [23|29845| + ], [24|29857| y ], [25|253| 0 ]

Arvore abstrata: [0 | 0 | m], [1 | 1 | =], [2 | 2 | r], [3 | 5 | 0], [4 | 3 | h], [5 | 4 | +], [6 | 9 | x], [7 | 10 | y]

**m(){ (1-1); r(1); }**

Palavra 3: m(){ (1-1); r(1); }

Palavra aceita.

Producoes: P1, P6, P21, P8, P13, P8, P8

Arvore sintatica: [0|0| S ], [1|1| M ], [2|13| m ], [3|14| ( ], [4|15| ) ], [5|16| { ], [6|17| C ],  
[7|18| ; ], [8|19| r ], [9|20| ( ], [10|21| E ], [11|22| ) ], [12|23| ; ], [13|24| } ], [14|205| ( ],  
[15|206| E ], [16|207| X ], [17|208| E ], [18|209| ) ], [19|2473| 1 ], [20|2485| - ],  
[21|2497| 1 ], [22|253| 1 ]

Arvore abstrata: [0 | 0 | m], [1 | 1 | -], [2 | 2 | r], [3 | 5 | 1], [4 | 3 | 1], [5 | 4 | 1]

**m(){ w(1) { (1/x); }; r(1); }**

Palavra 4: m(){ w(1) { (1/x); }; r(1); }

Palavra aceita.

Producoes: P1, P6, P22, P8, P21, P8, P15, P9, P8

Arvore sintatica: [0|0| S ], [1|1| M ], [2|13| m ], [3|14| ( ], [4|15| ) ], [5|16| { ], [6|17| C ],  
[7|18| ; ], [8|19| r ], [9|20| ( ], [10|21| E ], [11|22| ) ], [12|23| ; ], [13|24| } ], [14|205| w ],  
[15|206| ( ], [16|207| E ], [17|208| ) ], [18|209| { ], [19|210| C ], [20|211| ; ], [21|212| } ],  
[22|2485| 1 ], [23|2521| ( ], [24|2522| E ], [25|2523| X ], [26|2524| E ], [27|2525| ) ],  
[28|30265| 1 ], [29|30277| / ], [30|30289| x ], [31|253| 1 ]

Arvore abstrata: [0 | 0 | m], [1 | 1 | w], [2 | 2 | r], [3 | 5 | 1], [4 | 3 | 1], [5 | 4 | /], [6 | 9 | 1],  
[7 | 10 | x]

**$n() \{ (0/y); r(y); \}$   $g() \{ i=y; r(x); \}$   $m() \{ (1-x); r(0); \}$**

Palavra 5:  $n() \{ (0/y); r(y); \}$   $g() \{ i=y; r(x); \}$   $m() \{ (1-x); r(0); \}$

Palavra aceita.

Producoes: P3, P4, P21, P7, P15, P10, P10, P5, P17, P10, P9, P6, P21, P8, P13, P9, P7

Arvore sintatica: [0|0| S ], [1|1| N ], [2|2| G ], [3|3| M ], [4|13| n ], [5|14| ( ], [6|15| ) ],  
[7|16| { ], [8|17| C ], [9|18| ; ], [10|19| r ], [11|20| ( ], [12|21| E ], [13|22| ) ], [14|23| ; ],  
[15|24| } ], [16|205| ( ], [17|206| E ], [18|207| X ], [19|208| E ], [20|209| ) ], [21|2473| 0 ],  
[22|2485| / ], [23|2497| y ], [24|253| y ], [25|25| g ], [26|26| ( ], [27|27| ) ], [28|28| { ],  
[29|29| C ], [30|30| ; ], [31|31| r ], [32|32| ( ], [33|33| E ], [34|34| ) ], [35|35| ; ],  
[36|36| } ], [37|349| i ], [38|350| = ], [39|351| E ], [40|4213| y ], [41|397| x ], [42|37| m ],  
[43|38| ( ], [44|39| ) ], [45|40| { ], [46|41| C ], [47|42| ; ], [48|43| r ], [49|44| ( ],  
[50|45| E ], [51|46| ) ], [52|47| ; ], [53|48| } ], [54|493| ( ], [55|494| E ], [56|495| X ],  
[57|496| E ], [58|497| ) ], [59|5929| 1 ], [60|5941| - ], [61|5953| x ], [62|541| 0 ]

Arvore abstrata: [0 | 0 | S], [1 | 1 | n], [2 | 2 | G], [3 | 5 | g], [4 | 6 | m], [5 | 3 | /], [6 | 4 | r],  
[7 | 9 | y], [8 | 7 | 0], [9 | 8 | y], [10 | 11 | =], [11 | 12 | r], [12 | 25 | x], [13 | 23 | i],  
[14 | 24 | y], [15 | 13 | -], [16 | 14 | r], [17 | 29 | 0], [18 | 27 | 1], [19 | 28 | x]

**$m()\{w(x)\{f(y)\{k=(1+(1*0));\};\};r(0);\}$**

Palavra 6:  $m()\{w(x)\{f(y)\{k=(1+(1*0));\};\};r(0);\}$

Palavra aceita.

Producoes: P1, P6, P22, P9, P23, P10, P19, P11, P8, P12, P11, P8, P14, P7, P7

Arvore sintatica: [0|0| S ], [1|1| M ], [2|13| m ], [3|14| ( ], [4|15| ) ], [5|16| { ], [6|17| C ],  
[7|18| ; ], [8|19| r ], [9|20| ( ], [10|21| E ], [11|22| ) ], [12|23| ; ], [13|24| } ], [14|205| w ],  
[15|206| ( ], [16|207| E ], [17|208| ) ], [18|209| { ], [19|210| C ], [20|211| ; ], [21|212| } ],  
[22|2485| x ], [23|2521| f ], [24|2522| ( ], [25|2523| E ], [26|2524| ) ], [27|2525| { ],  
[28|2526| C ], [29|2527| ; ], [30|2528| } ], [31|30277| y ], [32|30313| k ], [33|30314| = ],  
[34|30315| E ], [35|363781| ( ], [36|363782| E ], [37|363783| X ], [38|363784| E ],  
[39|363785| ) ], [40|4365385| 1 ], [41|4365397| + ], [42|4365409| ( ], [43|4365410| E ],  
[44|4365411| X ], [45|4365412| E ], [46|4365413| ) ], [47|52384921| 1 ],  
[48|52384933| \* ], [49|52384945| 0 ], [50|253| 0 ]

Arvore abstrata: [0 | 0 | m], [1 | 1 | w], [2 | 2 | r], [3 | 5 | 0], [4 | 3 | x], [5 | 4 | f], [6 | 9 | y],  
[7 | 10 | =], [8 | 21 | k], [9 | 22 | +], [10 | 45 | 1], [11 | 46 | \*], [12 | 93 | 1], [13 | 94 | 0]



# Link para download

---

Código fonte e exemplos encontram-se para download no seguinte link:

<https://github.com/MatBrands/Compiladores/tree/master/Proj2/Proj2c>

# Referências

---

<https://web.stanford.edu/class/archive/cs/cs103/cs103.1156/tools/cfg/>