

## TRABALHO FINAL – parte 3: implementação do analisador sintático

Implementar o **analisador sintático** de forma que indique quais programas escritos na linguagem 2020.2 estão sintaticamente corretos, seguindo as orientações abaixo:

1º passo: efetue correções, se for o caso, na especificação dos *tokens* da linguagem conforme indicado na avaliação do analisador léxico.

2º passo: efetue correções, se for o caso, na gramática da linguagem conforme solicitado/indicado na avaliação do trabalho no.3.

3º passo: implemente o analisador sintático, bem como o tratamento de erros sintáticos, conforme especificado abaixo.

Entrada	– A entrada para o analisador sintático é um conjunto de caracteres, isto é, o programa fonte do editor do compilador.
Saída	<p>– Caso o botão <b>compilar</b> seja pressionado, a ação deve ser: executar as análises léxica e sintática do programa fonte e apresentar a saída. Um programa pode ser compilado com sucesso ou apresentar erros. Em cada uma das situações a saída deve ser:</p> <p><u>1ª situação:</u> programa compilado com sucesso</p> <p>✓ <b>mensagem</b> (<i>programa compilado com sucesso</i>), na área reservada para mensagens, indicando que o programa não apresenta erros.</p> <p>A <b>lista de tokens</b> <u>não deve mais</u> ser mostrada na área reservada para mensagens.</p> <p><u>2ª situação:</u> programa apresenta erros</p> <p>✓ <b>mensagem</b>, na área reservada para mensagens, indicando que o programa apresenta erro. O erro pode ser <b>léxico</b> ou <b>sintático</b>, cujas mensagens devem ser conforme descrito abaixo.</p> <p>As mensagens geradas pelo GALS devem ser alteradas.</p> <p>Caso seja pressionado o botão <b>compilar</b> e o editor não contenha nenhuma sequência de caracteres ou contenha apenas caracteres de formatação (espaço em branco, quebra de linha ou tabulação), deve ser apresentada a mensagem <i>nenhum programa para compilar</i> na área reservada para mensagens.</p>

### OBSERVAÇÕES:

- O tipo do analisador sintático a ser gerado é **LL (1)**.
- As mensagens para os **erros léxicos** devem ser conforme especificado na parte 2 do trabalho final.
- As mensagens para os **erros sintáticos** devem indicar a linha onde ocorreu o erro, o token encontrado (lexema) e o(s) símbolo(s) esperado(s), conforme explicado em aula e detalhado a seguir. Assim, tem-se alguns exemplos:

Erro na linha 1 – **encontrado** EOF **esperado** identificador

Erro na linha 1 – **encontrado** area **esperado** (

Observa-se que:

- para o não-terminal <lista\_de\_expressões>, ou com outro nome, usado para definir essa estrutura sintática especificada no trabalho no.2, a mensagem deve ser do tipo: **encontrado** ... **esperada** uma expressão
  - para todos os não-terminais alcançados a partir de <expressão> (inclusive), a mensagem deve ser do tipo: **encontrado** ... **esperada** uma expressão
  - quando for encontrado ou esperado fim de programa ou fim de arquivo ou \$, a mensagem deve ser encontrado (ou esperado) EOF
  - para os demais não-terminais, a mensagem deve ser do tipo: **encontrado** ... **esperado** símbolo<sub>1</sub>, símbolo<sub>2</sub>, ... símbolo<sub>n</sub>, conforme tabela de análise sintática, com as seguintes ressalvas:
    - ✓ quando for esperado cte\_int, cte\_float, cte\_str, true ou false, ao invés de listar todas as constantes, deve-se apresentar a mensagem **esperada** constante, observando que pode ser necessário complementar a mensagem com outros símbolos esperados, conforme exemplificado abaixo;
    - ✓ quando for esperado float, int ou str, ao invés de listar todos os tipos, deve-se apresentar a mensagem **esperado** tipo, observando que pode ser necessário complementar a mensagem com outros símbolos esperados, conforme exemplificado abaixo;
  - são exemplos de mensagens de erro **inadequadas**: <lista\_comandos> inválido, esperado cte\_int ou \$ inesperado
  - todas as mensagens de erro geradas pelo GALS devem ser mantidas (em comentário), MAS devem ser alteradas, conforme especificado acima.
- A gramática especificada no trabalho nº3 (com as devidas correções) deve ser usada para implementação do analisador sintático. Além disso, trabalhos desenvolvidos usando especificações diferentes daquelas elaboradas pela equipe no trabalho nº3 receberão nota 0.0 (zero).

Assim, por exemplo, considerando o seguinte “trecho” da tabela de análise sintática (menu Documentação > Tabela de Análise Sintática):

	id	cte_int	cte_float	cte_str	not	else	false	float	for	if	input	int	print	str	true	while	end	(	)	+	-
<programa>	0	-	-	-	-	-	-	-	0	0	-	-	0	-	-	0	-	-	-	-	-
<lista_comandos>	1	-	-	-	-	-	-	-	1	1	-	-	1	-	-	1	-	-	-	-	-
<atribuicao_entrada >	58	58	58	58	58	-	58	60	-	-	59	60	-	60	58	-	-	58	-	58	58
<tipo>	-	-	-	-	-	-	-	62	-	-	-	61	-	63	-	-	-	-	-	-	-
<lista_de_expressoes>	5	5	5	5	5	-	5	-	-	-	-	-	-	-	5	-	-	5	-	5	5
<expressao>	17	17	17	17	17	-	17	-	-	-	-	-	-	-	17	-	-	17	-	17	17

As mensagens de erro para os não-terminais relacionados devem ser:

- para o não-terminal <programa>: **encontrado ... esperado** id, for, if, print, while
- para o não-terminal <lista\_comandos>: **encontrado ... esperado** id, for, if, print, while
- para o não-terminal <atribuicao\_entrada>: **encontrado ... esperado** id, constante, not, tipo, input, (, +, -
- para o não-terminal <tipo>: **encontrado ... esperado** tipo
- para o não-terminal <lista\_de\_expressoes>: **encontrado ... esperada** uma expressão
- para o não-terminal <expressao>: **encontrado ... esperada** uma expressão
- A implementação do analisador sintático, bem como da interface do compilador e do analisador léxico, deve ser disponibilizada no AVA (na aba COMPILADOR), na “**pasta**” **da sua equipe**. Deve ser disponibilizado um **arquivo compactado** (com o nome: `sintatico`), contendo: o código fonte, o executável e o arquivo com as especificações léxica e sintática (no GALS, arquivo com extensão `.gals`).
- Na avaliação do analisador sintático serão levadas em consideração: a correta especificação da gramática, conforme trabalho nº3, a qualidade das mensagens de erro, conforme descrito acima e o uso apropriado de ferramentas para construção de compiladores. Observa-se que todas as mensagens de erro sintático geradas pelo GALS devem ser alteradas conforme especificado anteriormente.

**DATA:** entregar o trabalho até às 23h do dia 13/11/2020 (sexta-feira). Não serão aceitos trabalhos após data e hora determinados.

## EXEMPLOS DE ENTRADA / SAÍDA

**EXEMPLO 1:** com erro léxico

ENTRADA		SAÍDA (na área de mensagens)
linha		
1	lado = input ('Digite um valor para lado: )	Erro na linha 1 - constante str inválida ou não finalizada
2	area = lado * lado	
3	print area)	

**EXEMPLO 2:** com erro sintático

ENTRADA		SAÍDA (na área de mensagens)
linha		
1	lado = input ('Digite um valor para lado: ')	Erro na linha 3 - encontrado area esperado (
2	area = lado * lado	
3	print area)	

**EXEMPLO 3:** sem erro

ENTRADA		SAÍDA (na área de mensagens)
linha		
1	lado = input ('Digite um valor para lado: ')	programa compilado com sucesso
2	area = lado * lado	
3	print (area)	