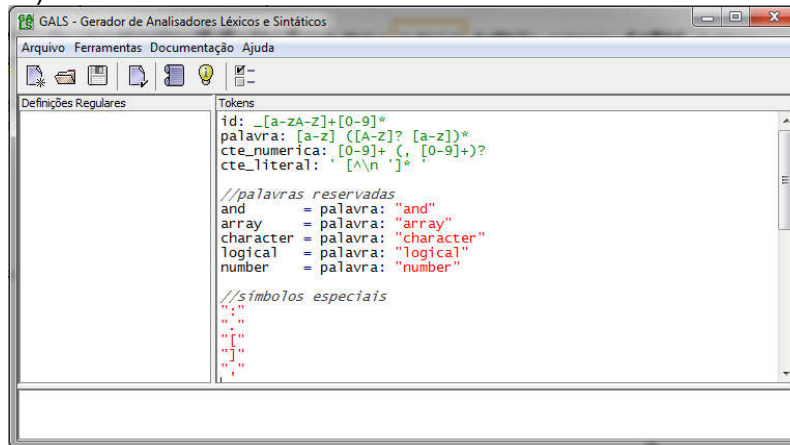


TRABALHO nº3: preparando a gramática 2018.1 para implementação do analisador sintático

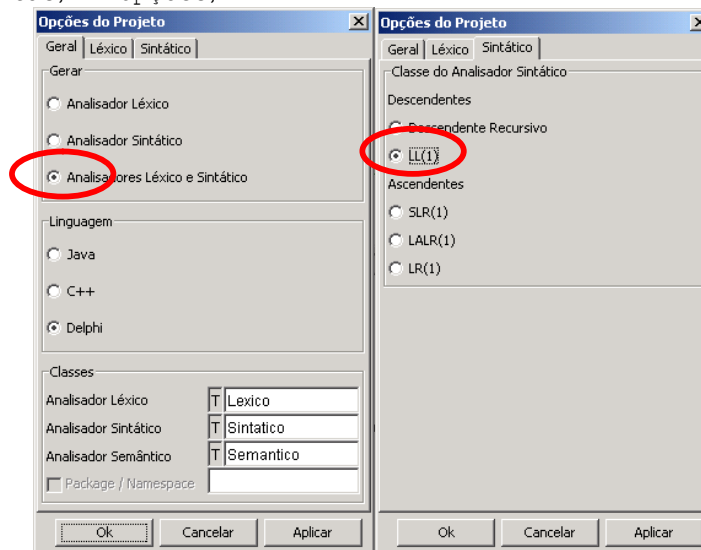
Considere a gramática especificada no trabalho nº2 (com as devidas correções) e os *tokens* especificados no trabalho nº1 (arquivo com especificações léxicas).

Siga as orientações abaixo (**dúvidas?** Ver exemplo de gramática em: <http://gals.sourceforge.net/help.html>)

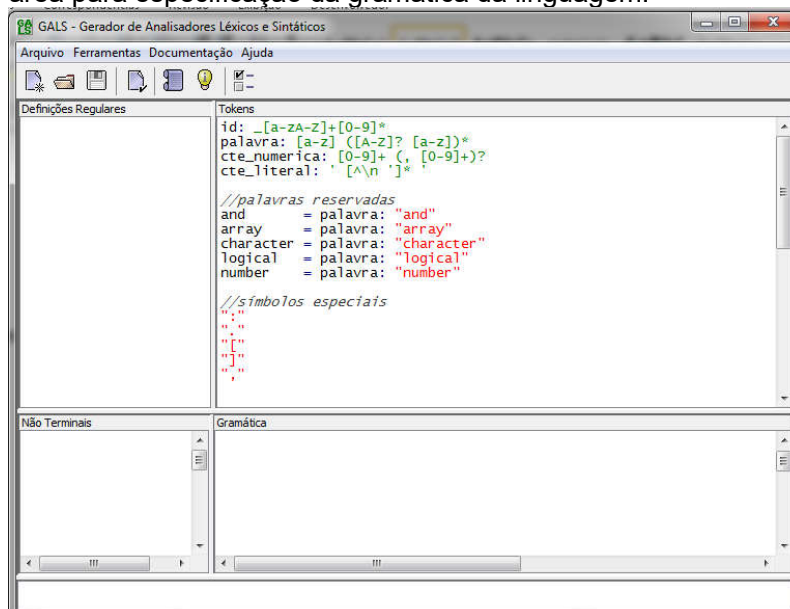
1º passo: abra o arquivo que contém a especificação dos *tokens* da linguagem, como no exemplo abaixo (de uma linguagem qualquer):



2º passo: no menu Ferramentas, em Opções, selecione:



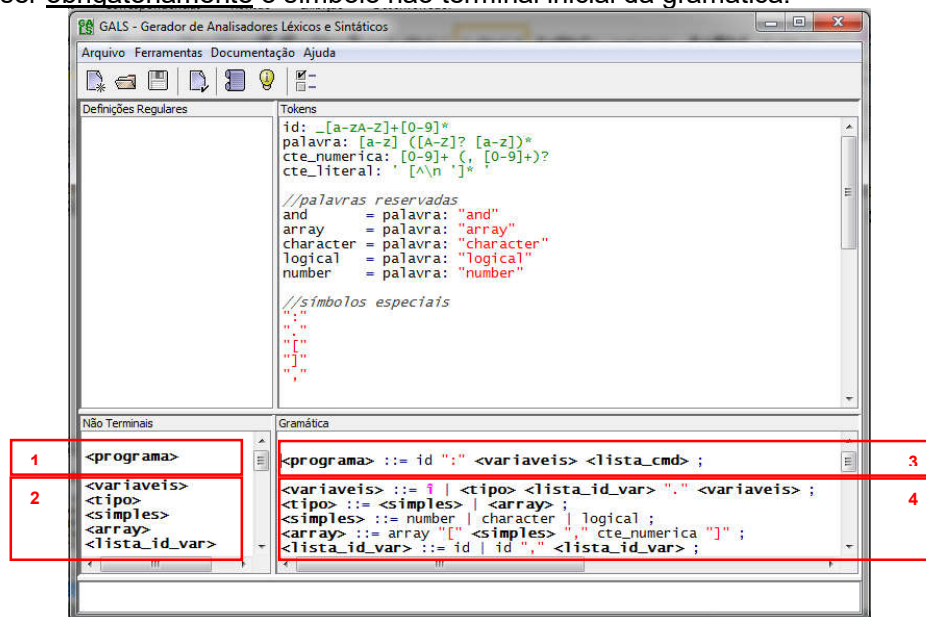
Será disponibilizada a área para especificação da gramática da linguagem:



3º passo: digite a gramática especificada no trabalho nº2 (com as devidas correções) na área Gramática utilizando a seguinte notação:

- **símbolos não terminais:** seguem o padrão de formação: `<[a-zA-Z0-9_]*>`
- **símbolos terminais:** são os *tokens* que devem ser escritos da seguinte forma:
 - ✓ os símbolos especiais devem estar entre aspas duplas
 - ✓ ϵ é representado por `ε` (i minúsculo com acento circunflexo)
 - ✓ as palavras reservadas definidas como: `array = palavra: "array"` devem ser colocadas na gramática sem aspas
 - ✓ os demais *tokens* devem ser escritos conforme especificado, isto é, se a constante numérica foi especificada como: `cte_numerica`, na gramática deve ser usada da mesma forma
- **regras sintáticas:** seguem o formato: `<não_terminal> ::= <lista_símbolos> | <lista_símbolos> | ...` onde `<lista_símbolos>` só pode conter símbolos não terminais e terminais já declarados, e a definição das regras sintáticas de um não-terminal deve ser finalizada com ponto e vírgula.

4º passo: digite os não terminais da gramática na área Não Terminais, sendo que o primeiro não terminal dessa lista deve ser **obrigatoriamente** o símbolo não terminal inicial da gramática.



Na figura acima temos que:

- item 1: não terminal inicial da gramática é o primeiro não terminal da lista de Não terminais
- item 2: demais não terminais da gramática em qualquer ordem
- item 3: regra sintática inicial da gramática, onde `id` é o identificador, conforme foi especificado nos *tokens*, `;"` é um símbolo especial, ponto e vírgula (`;`) é usado para indicar que as regras do não terminal `<programa>` estão completamente especificadas (no caso só uma)
- item 4: regras dos demais não terminais, onde no não terminal `<variaveis>` temos que `ε` é o ϵ , no não terminal `<array>` é usada a palavra reservada `array`, além dos *tokens* `"["`, `" "`, `"]"` e `cte_numerica`.

5º passo: verifique se a gramática está corretamente digitada selecionando `Verificar erros` no menu `Ferramentas` e, se for o caso, corrija os erros indicados.

6º passo: digite, após a última regra sintática da gramática, as regras sintáticas para o não terminal **<expressão>** (aritmética, lógica e relacional) conforme especificado abaixo, apenas alterando os identificadores dos *tokens* para os correspondentes determinados pela equipe na especificação léxica:

<code><expressão></code>	<code>::= <expressão> "&&" <elemento> <expressão> " " <elemento> <elemento></code>
<code><elemento></code>	<code>::= <relacional> true false "!" <elemento></code>
<code><relacional></code>	<code>::= <aritmética> <operador_relacional> <aritmética> <aritmética></code>
<code><operador_relacional></code>	<code>::= "=" "!=" "<" "<=" ">" ">="</code>
<code><aritmética></code>	<code>::= <aritmética> "+" <termo> <aritmética> "-" <termo> <termo></code>
<code><termo></code>	<code>::= <termo> "*" <fator> <termo> "/" <fator> <fator></code>
<code><fator></code>	<code>::= identificador identificador get "(" identificador ")" cte_inteira cte_real cte_caractere "(" <expressão> ")" "+" <fator> "-" <fator></code>

7º passo: verifique se a gramática está corretamente digitada selecionando `Verificar erros` no menu `Ferramentas` e, se for o caso, corrija os erros indicados.

8º passo: estando a gramática corretamente digitada, elimine recursão à esquerda (direta) usando o algoritmo apresentado em aula.

9º passo: elimine não determinismo à esquerda (direto e indireto) usando o algoritmo apresentado em aula.

10º passo: elimine os símbolos inúteis usando o algoritmo apresentado em aula.

OBSERVAÇÕES:

- As especificações feitas no trabalho nº2 (corrigidas) devem usadas para resolução deste trabalho. Observa-se que essas especificações devem ser adaptadas à notação da ferramenta para construção de compiladores utilizada (GALS ou outra). Além disso, trabalhos desenvolvidos usando especificações diferentes daquelas elaboradas pela equipe no trabalho nº2 receberão nota 0.0 (zero).
- O arquivo com as especificações léxicas e sintáticas – no GALS, arquivo com extensão `.gals` – deve ser disponibilizado no AVA na pasta da sua equipe.

DATA: entregar o trabalho até às 23h do dia **21/05/2018 (segunda-feira)**.