

1. Este trabalho é continuação do analisador léxico?

Sim! O analisador sintático é a segunda etapa do seu projeto de compilador.

Você deve aproveitar o analisador léxico que já desenvolveu. O sintático irá receber como entrada a lista de tokens produzida pelo seu próprio léxico, não mais o texto original do código-fonte.

2. Qual é a diferença prática entre analisador léxico e sintático neste projeto?

O léxico transforma o texto em tokens.

O sintático verifica se esses tokens, juntos, formam um programa válido conforme as regras da linguagem do seu projeto.

- **Analisador Léxico:**

Foi responsável por ler o código-fonte caractere por caractere, identificar e classificar tokens (palavras-chave, identificadores, operadores, números, etc.), além de detectar erros de formação de tokens.

Resultado: uma lista de tokens, cada um com tipo, lexema, linha e coluna.

- **Analisador Sintático:**

Agora, o seu trabalho é usar essa lista de tokens (resultado do léxico) para verificar se a sequência faz sentido de acordo com as regras da sua linguagem (a gramática).

O sintático detecta erros de estrutura, como comandos incompletos, blocos não fechados, ordem incorreta de comandos, etc.

Resultado: validação da estrutura do programa e mensagens de erro sintático, mostrando onde e por que o código está errado segundo a gramática criada por vocês.

3. Como implementar o analisador sintático aproveitando o léxico feito?

- O seu analisador sintático deve receber a lista de tokens gerada pelo léxico do próprio projeto do já feito.
- Não é necessário ler o código-fonte diretamente, mas sim trabalhar com os tokens já identificados.
- Sua implementação pode ser manual ou baseada em ferramentas/bibliotecas, desde que esteja bem explicada no PDF.

4. Preciso mostrar a árvore sintática na interface?

- Não é obrigatório, mas se conseguir mostrar (mesmo de forma simples, como uma tabela ou lista de regras reconhecidas), você pode ser avaliado.
- O principal é relatar claramente os erros sintáticos e onde eles ocorrem.

5. Que tipos de erros sintáticos devo detectar?

- Exemplos:
 - Parênteses, colchetes ou chaves abertos e não fechados
 - Comandos incompletos ou fora de ordem da gramática
 - Uso incorreto de operadores ou palavras-chave
 - Outros erros de estrutura, conforme a gramática definida pelo grupo

6. O que precisa ter na interface gráfica?

- **No mínimo:**
 - Abrir arquivo
 - Salvar arquivo
 - Listar tokens
 - Limpar
 - Executar análise sintática
 - Sair do programa
- Outras funções criativas também serão avaliadas e valorizadas (ex: mostrar erros destacados, árvore sintática etc.)

7. Como documentar as regras sintáticas (gramática)?

- No PDF, coloque as regras sintáticas que sua dupla usou, seja em formato formal ou em texto explicando como as frases da sua linguagem devem ser estruturadas.
- É importante deixar claro para que possamos entender o funcionamento do seu sintático.

8. O que devo entregar?

- Um arquivo .zip com todo o código-fonte do analisador sintático (continuação do seu projeto).
- Um arquivo .txt ou .pdf explicando:
 - Qual IDE e linguagem foi usada
 - Como executar o projeto (passo a passo)
 - As regras gramaticais usadas
 - Os principais erros detectados pelo analisador sintático
 - Outras informações que achar importante

9. Funcionalidades extras valem pontos?

- Sim! Criatividade, melhorias na interface, integração clara com o analisador léxico já feito, visualização da árvore sintática, entre outros, serão pontuados e avaliados.

Caso ainda tenha mais dúvidas, entre em contato!