Projeto Conceitual da Implementação do Compilador

Equipe: EQ03

Componentes da Equipe:

- William Franco Abdul Hai
- Nathalia Ohana Barigchum Leite
- Daniel Telles de Oliveira
- Pedro Felipe de Oliveira Facundes

Introdução

Este documento apresenta o projeto conceitual da implementação do compilador RainCheck, um Static Checker desenvolvido para a linguagem de programação Storm2024-2. O projeto faz parte da disciplina de Compiladores do curso de Engenharia da Computação do SENAI CIMATEC.

Objetivos do Projeto

O objetivo principal do projeto é desenvolver um Static Checker que seja capaz de:

- Realizar a análise léxica completa de arquivos fonte escritos em Storm2024-2.
- Identificar e classificar corretamente todos os tokens definidos pela linguagem.
- Construir uma tabela de símbolos eficiente, armazenando identificadores e seus atributos.
- Gerar relatórios .LEX e .TAB formatados conforme especificações, auxiliando na validação e depuração de códigos fonte.
- Preparar a base para futuras implementações de análise sintática e semântica da linguagem.

Descrição dos Componentes Principais

Leitura do Arquivo

Responsabilidades:

- Receber o nome do arquivo fonte como parâmetro de entrada.
- Verificar e ajustar a extensão do arquivo para .242 quando necessário.
- Abrir o arquivo para leitura e tratar possíveis erros.
- Fornecer os caracteres do arquivo para o analisador léxico.

Analisador Léxico

Responsabilidades:

- Ler o arquivo fonte caractere a caractere.
- Identificar tokens válidos conforme os padrões léxicos da linguagem Storm2024-2.
- Ignorar espaços em branco, tabulações, quebras de linha e comentários.
- Tratar a insensibilidade a maiúsculas e minúsculas.
- Aplicar o limite de 30 caracteres válidos para identificadores.
- Armazenar tokens identificados e seus atributos.

Tabela de Símbolos

Responsabilidades:

- Armazenar identificadores únicos encontrados durante a análise léxica.
- Registrar atributos de cada símbolo:
    - Número da entrada.
    - Código do átomo.
    - Lexema.
    - Quantidade de caracteres antes e depois da truncagem.
    - Tipo do símbolo.
    - Linhas de ocorrência.

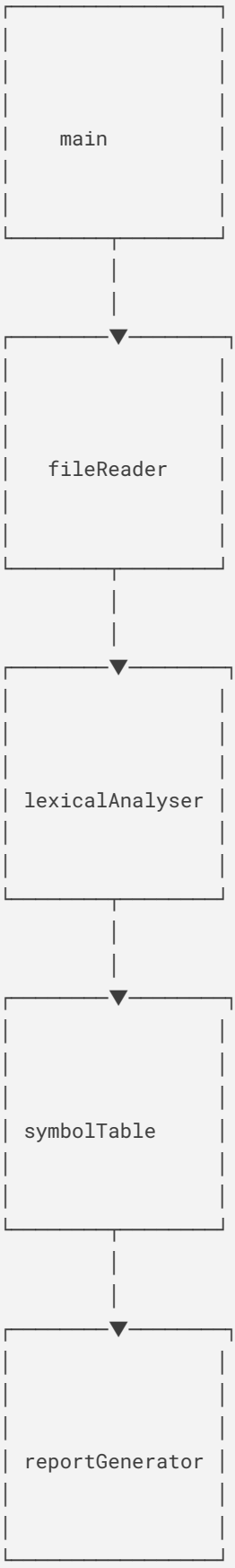
Gerador de Relatórios

Responsabilidades:

- Gerar o arquivo .LEX com o relatório da análise léxica.
- Gerar o arquivo .TAB com o relatório da tabela de símbolos.
- Formatar os relatórios conforme especificações, incluindo cabeçalhos com informações da equipe e do arquivo analisado.

Diagrama de Estrutura

- Main: Controla o fluxo principal do programa, interagindo com os módulos.
- FileReader: Gerencia a leitura do arquivo fonte.
- LexicalAnalyzer: Analisa caracteres e identifica tokens.
- SymbolTable: Armazena identificadores e atributos.
- ReportGenerator: Gera os relatórios .LEX e .TAB.



Especificação dos Átomos e Atributos

Lista de Átomos Suportados

Palavras-chave (A01 - A26):

Átomo	Cód	Descrição
cadeia	A01	Tipo de dado string
caracter	A02	Tipo de dado caractere
declaracoes	A03	Início das declarações
enquanto	A04	Estrutura de repetição “enquanto”
false	A05	Valor booleano falso
fimDeclaracoes	A06	Fim das declarações
fimEnquanto	A07	Fim da estrutura “enquanto”
fimFunc	A08	Fim de uma função
fimFuncoes	A09	Fim do bloco de funções
fimPrograma	A10	Fim do programa
fimSe	A11	Fim da estrutura “se”
funcoes	A12	Bloco de funções
imprime	A13	Comando para imprimir saída
inteiro	A14	Tipo de dado inteiro
logico	A15	Tipo de dado booleano
pausa	A16	Comando para pausar execução
programa	A17	Início do programa
real	A18	Tipo de dado real (decimal)
retorna	A19	Comando para retornar valor
se	A20	Estrutura condicional “se”
senao	A21	Estrutura condicional “senão”
tipoFunc	A22	Definição de tipo de função
tipoParam	A23	Tipo de parâmetro
tipoVar	A24	Tipo de variável
true	A25	Valor booleano verdadeiro
vazio	A26	Tipo de dado vazio (void)

Operadores (B01 - B22)

Átomo	Cód	Descrição
%	B01	Operador de módulo
(	B02	Abre parênteses
)	B03	Fecha parênteses
,	B04	Separador de argumentos
:	B05	Símbolo de dois pontos
:=	B06	Operador de atribuição
;	B07	Terminador de instrução
?	B08	Operador condicional ternário
[	B09	Abre colchetes
]	B10	Fecha colchetes
{	B11	Abre chaves
}	B12	Fecha chaves
-	B13	Operador de subtração
*	B14	Operador de multiplicação
/	B15	Operador de divisão
+	B16	Operador de adição
!=	B17	Operador de desigualdade
#	B17	Símbolo para comentários
<	B18	Operador “menor que”
<=	B19	Operador “menor ou igual a”
==	B20	Operador de igualdade
>	B21	Operador “maior que”
>=	B22	Operador “maior ou igual a”

Variáveis e Funções (C01 - C07)

Átomo	Cód	Descrição
consCadeia	C01	Constante do tipo string
consCaracter	C02	Constante do tipo caractere
consInteiro	C03	Constante do tipo inteiro

Átomo	Cód	Descrição
consReal	C04	Constante do tipo real (decimal)
nomFuncao	C05	Nome de uma função
nomPrograma	C06	Nome de um programa
variavel	C07	Nome de uma variável

Submáquinas (D0n)

Átomo	Cód	Descrição
subMáquinan	D0n	Submáquina n

Atributos da Tabela de Símbolos

- Número da Entrada: Índice único na tabela.
- Código do Átomo: Conforme codificação acima.
- Lexema: Nome do identificador, até 30 caracteres.
- Quantidade de Caracteres Antes da Truncagem: Tamanho original.
- Quantidade de Caracteres Depois da Truncagem: Tamanho armazenado.
- Tipo do Símbolo:
    - Tipos de Dados: PFO (ponto flutuante), INT (inteiro), STR (string), CHC (caractere), BOO (booleano), VOI (void).
    - Arrays Correspondentes: APF, AIN, AST, ACH, ABO.
- Linhas de Ocorrência: Até as cinco primeiras linhas.

Pseudo-Código do Analisador Léxico

```
INICIAR AnalisadorLexico
    Abrir arquivo fonte para leitura
    Se erro na abertura do arquivo
        Exibir mensagem de erro e encerrar
    Converter conteúdo para maiúsculas
    Inicializar:
        posição <- 0
        linha <- 1
        coluna <- 1
        listaTokens <- []
        tabelaSimbolos <- inicializarTabela()
    Enquanto não fim do arquivo
        caractere <- lerCaractere()
        Se caractere é espaço, tabulação ou nova linha
            Atualizar linha e coluna
            Continuar
        Se início de comentário
            Ignorar comentário
            Continuar
        Se letra ou '_'
            Montar identificador ou palavra-chave
            Aplicar limite de 30 caracteres
            Se palavra-chave
                Registrar token
            Senão
                Registrar token
                Atualizar tabela de símbolos
        Se dígito
            Montar número (inteiro ou real)
            Registrar token
        Se '"' ou '''
            Montar constante de cadeia ou caractere
            Registrar token
        Se operador ou delimitador
            Verificar operadores de múltiplos caracteres
            Registrar token
        Senão
            Ignorar caractere inválido
    Fechar arquivo
    Retornar listaTokens e tabelaSimbolos
FIM
```

Planejamento de Implementação

Principais Funções e Módulos

- main
  - Processar argumentos de entrada.
  - Controlar fluxo geral.
- fileReader
  - Abrir e ler arquivo fonte.
  - Fornecer caracteres para o analisador léxico.
- lexicalAnalyser

    - Identificar e classificar tokens.
    - Interagir com a tabela de símbolos.
- symbolTable
    - Armazenar e gerenciar identificadores.
- reportGenerator
    - Gerar arquivos .LEX e .TAB.

Fluxo de Execução Geral

1. Inicialização
   a) O programa é iniciado e processa os argumentos.
   b) Verifica e ajusta o nome do arquivo.
2. Leitura do Arquivo
   a) O FileReader abre o arquivo e prepara para leitura.
3. Análise Léxica
   a) O LexicalAnalyzer lê caracteres e identifica tokens.
   b) Atualiza a SymbolTable conforme necessário.
4. Geração de Relatórios
   a) O ReportGenerator gera os arquivos .LEX e .TAB.
5. Finalização
   a) Recursos são liberados e o programa é encerrado.

Listagem de Dúvidas

1. Como tratar comentários não fechados no final do arquivo?
   a) Devemos considerar todo o restante como comentário ou reportar um erro?
2. Se um identificador ultrapassar 30 caracteres, devemos emitir um aviso?
   a) A especificação não esclarece se devemos notificar sobre truncagem.
3. Como proceder com caracteres inválidos dentro de um identificador?
   a) Devemos ignorá-los ou interromper o identificador?
4. Em casos de números malformados, qual é o comportamento esperado?
   a) Precisamos de orientação sobre como lidar com essas situações.