CES41 – Compiladores

Análise Sintática da linguagem COMP-ITA 2019

Laboratório 3
16 de Abril de 2019

Aluno: Felipe Vieira Coimbra
Professor: Fábio Carneiro Mokarzel

Resultados Obtidos

O analisador léxico foi concluído com sucesso, a entrada usada como teste e a saída correspondente podem ser vistas no arquivo **teste.in** e **teste.out** respectivamente.
O analisador sintático foi feito com sucesso, sendo capaz de analisar corretamente a sintaxe de um programa escrito na linguagem COMP-ITA 2019.

A entrada utilizada para teste foi:

```
program AnaliseDeTexto { global: char nomes[50,10], palavra[10]; int ntab, nocorr[50]; char c; logic fim; functions: int Procura () { local: int i, inf, sup, med, posic, compara; logic achou, fimteste; statements: achou <- false; inf <- 1; sup <- ntab; while (!achou && sup >= inf) { med <- (inf + sup) / 2; compara <- 0; fimteste <- false; for (i <- 0; !fimteste && compara = 0; i <- i+1) { if (palavra[i] < nomes[med,i]) compara <- ~1; else if (palavra[i] > nomes[med,i]) compara <- 1; if (palavra[i] = '\0' || nomes[med,i] = '\0') fimteste <- true; } if (compara = 0) achou <- true; else if (compara < 0) sup <- med - 1; else inf <- med + 1; } if (achou) posic <- med; else posic <- ~inf; return posic; } void Inserir (int posic) { local: int i, j; logic fim; statements: ntab <- ntab + 1; for (i <- ntab; i >= posic+1; i <- i-1) { fim <- false; for (j <- 0; !fim; j <- j+1) { nomes[i,j] <- nomes[i-1,j]; if (nomes[i,j] = '\0') fim <- true; } nocorr[i] <- nocorr[i-1]; } fim <- false; for (j <- 0; !fim; j <- j+1) { nomes[posic,j] <- palavra[j]; if (palavra[j] = '\0') fim <- true; } nocorr[posic] <- 1; } void ExibirTabela () { local: int i; logic fim; statements: write ("          ", "Palavra          ", "  Num. de ocorr."); for (i <- 1; i <= 50; i <- i+1) write ("-"); for (i <- 1; i <= ntab; i <- i+1) { write ("\n          "); fim <- false; for (j <- 0; !fim; j <- j+1) { if (nomes[i,j] = '\0') fim <- true; else write (nomes[i,j]); } write (" | ", nocorr[i]); } } main { local: int i, posic; char c; logic fim; statements: ntab <- 0; write ("Nova palavra? (s/n): "); read (c); while (c = 's' || c = 'S') { write ("\nDigite a palavra: "); fim <- false; for (i <- 0; !fim; i <- i+1) { read (palavra[i]); if (palavra[i] = '\n') { fim <- true; palavra[i] <- '\0'; } } posic <- Procura (); if (posic > 0) nocorr[posic] <- nocorr[posic] + 1; else call Inserir (~posic, i); write ("\n\nNova palavra? (s/n): "); read (c); } call ExibirTabela (); } }
```

Esse exemplo trata-se do programa exemplo fornecido retirado todas as quebras de linhas, tabulações e espaços desnecessários.

A saída gerada foi:

```
program AnaliseDeTexto {

global:
	char nomes[50, 10], palavra[10];
	int ntab, nocorr[50];
	char c;
	logic fim;

functions:
	int Procura() {
	local:
		int i, inf, sup, med, posic, compara;
		logic achou, fimteste;
	statements:
		achou <- false;
		inf <- 1;
		sup <- ntab;
		while (!achou && sup >= inf)
		{
			med <- (inf + sup)/2;
			compara <- 0;
			fimteste <- false;
			for (i <- 0; !fimteste && compara = 0; i <- i + 1)
			{
				if (palavra[i] < nomes[med, i])
					compara <- ~1;
				else
					if (palavra[i] > nomes[med, i])
						compara <- 1;
				if (palavra[i] = '\0' || nomes[med, i] = '\0')
					fimteste <- true;
			}
			if (compara = 0)
				achou <- true;
			else
				if (compara < 0)
					sup <- med - 1;
				else
					inf <- med + 1;
		}
		if (achou)
			posic <- med;
		else
```

```
                posic <- ~inf;
        return posic;

}

void Inserir(int posic) {
local:
        int i, j;
        logic fim;
statements:
        ntab <- ntab + 1;
        for (i <- ntab; i >= posic + 1; i <- i - 1)
        {
                fim <- false;
                for (j <- 0; !fim; j <- j + 1)
                {
                        nomes[i, j] <- nomes[i - 1, j];
                        if (nomes[i, j] = '\0')
                                fim <- true;
                }
                nocorr[i] <- nocorr[i - 1];
        }
        fim <- false;
        for (j <- 0; !fim; j <- j + 1)
        {
                nomes[posic, j] <- palavra[j];
                if (palavra[j] = '\0')
                        fim <- true;
        }
        nocorr[posic] <- 1;
}

void ExibirTabela() {
local:
        int i;
        logic fim;
statements:
        write("          ", "Palavra             ", "   Num. de ocorr.");
        for (i <- 1; i <= 50; i <- i + 1)
                write("-");
        for (i <- 1; i <= ntab; i <- i + 1)
        {
                write("\n          ");
                fim <- false;
                for (j <- 0; !fim; j <- j + 1)
                {
                        if (nomes[i, j] = '\0')
                                fim <- true;
```

```
                        else
                                write(nomes[i, j]);
                }
                write(" | ", nocorr[i]);
        }
}

main {
        local:
                int i, posic;
                char c;
                logic fim;
        statements:
                ntab <- 0;
                write("Nova palavra? (s/n): ");
                read(c);
                while (c = 's' || c = 'S')
                {
                        write("\nDigite a palavra: ");
                        fim <- false;
                        for (i <- 0; !fim; i <- i + 1)
                        {
                                read(palavra[i]);
                                if (palavra[i] = '\n')
                                {
                                        fim <- true;
                                        palavra[i] <- '\0';
                                }
                        }
                        posic <- Procura();
                        if (posic > 0)
                                nocorr[posic] <- nocorr[posic] + 1;
                        else
                                call Inserir(~posic, i);
                        write("\n\nNova palavra? (s/n): ");
                        read(c);
                }
                call ExibirTabela();
        }

}
```