

Um Compilador para a Linguagem TIGER usando a Linguagem JAVA como Ferramenta de desenvolvimento

POC II

- Aluno: Gustavo Brandão Escalda
- Orientadora: Mariza Andrade da Silva Bigonha

Trabalho Desenvolvido

- Implementação de um compilador para a linguagem TIGER em JAVA

Objetivos

- Contribuir para com o ensino da linguagem JAVA no curso de Compiladores
- Auxiliar na formação do aluno em um ambiente de programação de grande interesse

Linguagem TIGER

- Funções aninhadas
- Arranjos
- Variáveis de tipo inteiro e string
- Construções de controle estruturadas

Exemplo de programa em TIGER

```
/* Exemplo de um programa em TIGER que  
   calcula a soma de dois inteiros */  
let  
    var i : int := 0  
    var j : int := 5  
in  
    i := i + j  
end
```

Etapas do Compilador

- Análise Léxica
- Análise Sintática
- Análise Semântica
- Tradução para código intermediário

Análise Léxica

- Agrupamento de seqüências de caracteres em tokens

- JLex

- Exemplo:

<u>LEXEMA</u>	<u>TOKEN</u>
let	LET
var	VAR
i	ID
:	DOISPONTOS
int	ID
:=	ATRIB
0	INT
...	...
in	IN
i	ID
:=	ATRIB
i	ID
+	MAIS
j	ID
end	END

Análise Sintática

- Agrupamento de tokens em frases gramaticais
- CUP - LALR Parser Generator for JAVA
- Exemplo:

ERRADO

```
let var x: int  
in  
end
```

CERTO

```
let var x: int:= 0  
in  
end
```


Árvore Abstrata

- Exprime a estrutura das expressões do programa fonte omitindo detalhes irrelevantes da sintaxe concreta

- Exemplo:

```
LetExp(  
  DecList(  
    VarDec(i,  
      int,  
      IntExp(0)),  
    DecList(  
      VarDec(j,  
        int,  
        IntExp(5)),  
      DecList()),  
  SeqExp(  
    ExpList(  
      AssignExp(  
        SimpleVar(i),  
        OpExp(  
          PLUS,  
          varExp(  
            SimpleVar(i)),  
            varExp(  
              SimpleVar(j)))))
```

Análise Semântica

- Verificação de erros semânticos
- Captura de informações de tipo
- Exemplo semanticamente incorreto:

```
let
  type tipoarranjo = array of int
  var Arranjo:tipoarranjo := tipoarranjo
  [10] of 0
in
  Arranjo[2] = "nome"
end
```

Tradução para Código Intermediário

- Geração de código de baixo nível não atrelado a nenhuma linguagem ou máquina específica
- Exemplo:

```
null:
SEQ (
  SEQ (
    MOVE (
      TEMP t0,
      CONST 0),
    MOVE (
      TEMP t1,
      CONST 5)),
  MOVE (
    TEMP t0,
    BINOP (PLUS,
      TEMP t0,
      TEMP t1)))
```

Conclusão

- Construção do Front-end do compilador
- Disponibilização de uma nova ferramenta para o curso de Compiladores