

Mariza Andrade da Silva Bigonha

SIC: SISTEMA DE IMPLEMENTAÇÃO DE COMPILADORES

Tese apresentada ao Instituto de
Ciências Exatas da Universidade
Federal de Minas Gerais, como
requisito parcial para a obtenção do
título de Mestre em Ciência da
Computação.

Belo Horizonte
1985

Ao Roberto
A Patricia
Aos meus pais

AGRADECIMENTOS

Ao Prof. Roberto da Silva Bigonha pela orientação acadêmica desta tese e pela própria existência do Sistema SIC. Sua competência, seu entusiasmo e sua sistemática de trabalho muito me ensinaram.

Ao CNPq pelo apoio financeiro.

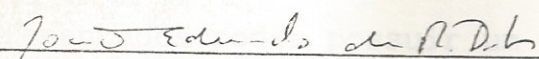
A todos aqueles que, de alguma forma, colaboraram na execução deste trabalho.

E acima de tudo, eu agradeço ao Roberto, meu amigo, companheiro e esposo, que através de seu amor, compreensão e paciência, me apoiou durante toda a execução deste trabalho.

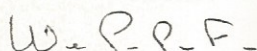
"SIC: SISTEMA DE IMPLEMENTAÇÃO DE COMPILADORES"

Mariza Andrade da Silva Bigonha

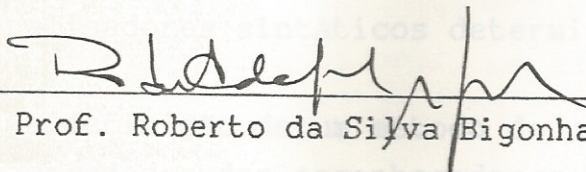
Tese defendida e aprovada pela banca examinadora constituída dos senhores:



Prof. João Eduardo Rezende Dantas



Prof. Wilson Pádua Paula Filho



Orientador:

Prof. Roberto da Silva Bigonha

DEPARTAMENTO DE CIENCIA DA COMPUTAÇÃO DO ICEx

Belo Horizonte, 5 de julho de 1985.

SINOPSE

Este trabalho apresenta a especificação e implementação de uma ferramenta, denominada SIC, destinada à assistir a implementação de compiladores através de uma linguagem especial, denominada LPC, que é baseada no Pascal.

LPC é dotada de meios para especificar sintaxe, associar rotinas semânticas a produções da gramática, para definir explicitamente atributos e implementar compiladores de um ou mais passos operando diretamente no fonte, interativos ou não.

Os compiladores implementados podem possuir um recuperador de erros independente de linguagem que fornece mensagens de erro automaticamente, com possibilidade de substituição ou inserção de não-terminais na entrada ou na pilha.

O Sistema ainda provê facilidades que permitem a resolução "ad-hoc" de conflitos da Tabela LALR(1) e o uso de gramáticas ambíguas na produção de analisadores sintáticos determinísticos.

E também apresentada a definição de um método de compactação de tabelas baseado em propriedades dos reconhecedores LR(1).

ABSTRACT

This dissertation presents the especification and implementation of a programming tool, called SIC, whose purpose is to assist compiler construction activities by means of a special purpose language (LPC) based on Pascal.

LPC provides facilities to specify the syntax of programming languages, associate semantic routines to grammar productions and allows explicit definition of attributes, facilities to implement interactive or batch compilers organized into one or more pass.

SIC implements an error handling and recovery routine which automatically issues error's messages, possibly inserting or substituting grammar symbols on the input and the stack.

The System also presents facilities to solve LALR(1) conflicts resulting from the use of ambiguous grammar. A method for table compression based on LR(1) properties has also been developed.

SUMARIO

pags.

1.	INTRODUÇÃO.....	1
1.1	MOTIVAÇÃO.....	1
1.2	CONTEUDO.....	2
2.	SISTEMA DE IMPLEMENTAÇÃO DE COMPILADORES (SIC).....	4
2.1	ORGANIZAÇÃO DO SISTEMA.....	4
2.2	EXECUÇÃO DO SIC.....	8
2.2.1	INTERAÇÃO COM O USUARIO.....	8
2.2.2	SAIDAS DO SIC.....	9
2.2.3	ARQUIVOS DO SIC.....	10
2.3	METODOLOGIA DE IMPLEMENTAÇÃO.....	11
2.4	LINGUAGEM DE IMPLEMENTAÇÃO DO SIC.....	12
3.	LINGUAGEM DE PROGRAMAÇÃO DE COMPILADORES (LPC).....	14
3.1	DEFINIÇÃO DA LPC.....	14
3.1.1	NOTAÇÃO.....	14
3.1.2	SIMBOLOS BASICOS.....	15
3.1.3	PROGRAMA EM LPC.....	19
3.1.3.1	OPÇÕES DE COMPILAÇÃO.....	20
3.1.3.2	PASSOS DO COMPILADOR.....	21
3.1.3.3	SEÇÃO UM.....	21
3.1.3.4	SEÇÃO DOIS.....	22
3.1.4	SEÇÃO DE TERMINAIS.....	22
3.1.5	SEÇÃO DE ATRIBUTOS.....	23
3.1.6	SEÇÃO DE ABRIDORES E FECHADORES DE ESCOPO	25
3.1.7	SEÇÃO MAPA DE NÃO-TERMINAIS.....	26
3.1.8	SEÇÃO DE DECLARAÇÕES.....	27
3.1.9	SEÇÃO GRAMATICA E ROTINAS SEMANTICAS.....	30
3.1.10	SEÇÃO RESOLUÇÃO DE CONFLITOS.....	34
3.1.11	CORPO.....	36
4	IMPLEMENTAÇÃO DA LPC.....	38
4.1	INTRODUÇÃO.....	38
4.2	ARQUIVOS.....	38
4.3	ESTRUTURAS DE DADO.....	41

4.3.1	TABELA DE SIMBOLOS DA GRAMATICA.....	41
4.3.2	TABELA DE ATRIBUTOS.....	43
4.3.3	TABELA DE PALAVRAS CHAVES.....	43
4.4	COMPILAÇÃO.....	44
4.4.1	OPÇÕES E PASSOS.....	44
4.4.2	TERMINAIS.....	45
4.4.3	DECLARAÇÕES.....	46
4.4.4	ATRIBUTOS.....	48
4.4.5	MAPA DE ABRIDORES E FECHADORES DE ESCOPO.....	53
4.4.6	MAPA DE NÃO-TERMINAIS.....	55
4.4.7	GRAMATICA E ROTINAS SEMANTICAS.....	55
4.4.8	RESOLUÇÃO DE CONFLITOS.....	60
4.4.9	CORPO DE UM PROGRAMA EM SIC.....	61
4.4.10	GERAÇÃO DOS IDENTIFICADORES PADROES.....	61
4.4.11	ANALISADORES LEXICOS DO SIC.....	63
5.	ANALISE SINTATICA E COMPACTAÇÃO DE TABELAS LR(1) EM SIC.....	68
5.1	INTRODUÇÃO.....	68
5.2	ESTRUTURA DO ANALISADOR LR(1).....	68
5.3	CODIFICAÇÃO DE TABELAS LR(1).....	71
5.3.1	TECNICAS DE REPRESENTAÇÃO DE MATRIZES ESPARSAS.....	71
5.3.2	METODO DE AHO & ULLMAN.....	71
5.4	O METODO PROPOSTO.....	74
5.5	EFEITOS NA RECUPERAÇÃO DE ERRO.....	80
5.6	NOVO ANALISADOR LR(1).....	81
5.7	IMPLEMENTAÇÃO.....	82
5.8	AValiação E CONCLUSÃO.....	86
6.	RECUPERAÇÃO DE ERROS SINTATICOS.....	88
6.1	OBJETIVO E ABORDAGEM ADOTADA.....	88
6.2	DESCRIÇÃO DO METODO.....	91
6.2.1	INTRODUÇÃO.....	91
6.2.2	AS TRES FASES DA RECUPERAÇÃO.....	91
6.2.3	PRIMEIRA FASE DE RECUPERAÇÃO.....	92

6.2.3.1	ESTRATEGIA 1:INSERÇÃO DE UM SIMBOLO.....	94
6.2.3.2	ESTRATEGIA 2:REMOÇÃO DE UM SIMBOLO.....	95
6.2.3.3	ESTRATEGIA 3:SUBSTITUIÇÃO DE UM SIMBOLO.....	96
6.2.4	SEGUNDA FASE DE RECUPERAÇÃO.....	97
6.2.5	TERCEIRA FASE DE RECUPERAÇÃO.....	98
6.3	CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO	100
6.3.1	NOVO ANALISADOR LR(1).....	100
6.3.2	ANALISADOR DE ERRO.....	103
6.3.3	TRANSIÇÃO "DEFAULT".....	106
6.3.4	ESCOLHA DE UM CANDIDATO PARA A CORREÇÃO.....	107
6.3.5	ORDEM DE PREFERENCIA ENTRE AS ESTRATEGIAS DE CORREÇÃO.....	108
6.3.6	USO DE MAPA DEPENDENTE DE LINGUAGEM.....	109
6.3.7	POSSIBILIDADE DE INSERÇÃO E SUBSTITUIÇÃO DE SIMBOLOS NÃO-TERMINAIS.....	109
6.4	EXEMPLOS DE RECUPERAÇÃO.....	110
7	COMPILADORES GERADOS PELO SIC.....	117
7.1	INTRODUÇÃO.....	117
7.2	COMPILADOR DE UM UNICO PASSO.....	117
7.3	COMPILADOR DE VARIOS PASSOS.....	118
7.4	COMPILAÇÃO INTERATIVA.....	119
8	CONCLUSÃO.....	122
	REFEFENCIAS BIBLIOGRAFICAS.....	124
	APENDICE I: DEFINIÇÃO SINTATICA DE LPC.....	127
	APENDICE II: GRAMATICA DA LINGUAGEM EXEMPLO.....	130
	APENDICE III	
III.1	EXEMPLO COMPLETO DA SAIDA DE UM COMPILADOR DE UM UNICO PASSO.....	131
III.2	SAIDA DA COMPILAÇÃO DO EXEMPLO DO APENDICE III.1...	157

III.3	EXEMPLO COMPLETO DO PRIMEIRO PASSO DE UM COMPILADOR DE VARIOS PASSOS.....	182
III.4	EXEMPLO COMPLETO DOS OUTROS PASSOS DE UM COMPILADOR DE VARIOS PASSOS.....	202
	APENDICE IV: LISTAGEM DOS VARIOS ANALISADORES SINTATICOS....	214
	APENDICE V: LISTAGEM DA IMPLEMENTAÇÃO DO SISTEMA SIC.....	284

LISTA DE FIGURAS

Figura 1: PRIMEIRO PASSO DO SIC.....	6
Figura 2: SEGUNDO PASSO DO SIC.....	7
Figura 3: TERCEIRO PASSO DO SIC.....	8
Figura 4: PROGRAMA LPC COMPILADO.....	8
Figura 5: MAPEAMENTO DE ABRIDORES E FECHADORES DE ESCOPO NOS ARQUIVOS FOPR e FCLSR RESPECTIVAMENTE.....	54
Figura 6: CONFIGURAÇÃO DO ARQUIVO GRAMSIC.....	57
Figura 7: CONFIGURAÇÃO DA PILHA IMEDIATAMENTE ANTES DE EXECUTAR A ROTINA SEMANTICA R1.....	59
Figura 8: RECONHECEDOR LR(1).....	69
Figura 9: TABELA LR(1).....	70
Figura 10: VETOR LR.....	77
Figura 11: CODIFICAÇÃO DA MATRIZ LR(1) DA FIGURA 9.....	79
Figura 12: NOVO ANALISADOR LR(1).....	81
Figura 13: NOVO ANALISADOR LR(1).....	101
Figura 14: CONFIGURAÇÃO DE BUF.....	103