UM GERADOR DE ANALISADORES SINTÁTICOS DE PRECEDÊNCIA

Benedito Ferreira de Oliveira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M. Sc)

Aprovada por:

Prof. José Lucas Mourão Rangel Netto

Prof. Guilherme Chagas Rodrigues

Prof. Luiz Fernando Loureiro Legey

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO - 1980

OLIVEIRA, BENEDITO FERREIRA DE

Um Gerador de Analisadores Sintáticos de Precedência (Rio de Janeiro) 1980

VII, 174p. 29,7 cm COPPE-UFRJ, M. Sc, Engenharia de Sistemas e Computação, 1980

Tese - Univ. Fed. do Rio de Janeiro. Faculdade de Engenharia

1. Linguagens Formais e Compiladores I.COPPE/UFRJ II. Um Gerador de Analisadores Sintáticos de Precedência (série)

Ao SERPRO e aos colegas de trabalho:

eles deixaram este trabalho acontecer.

RESUMO

A tese apresenta um sistema para projeto e análise de gramáticas e linguagens de precedência simples e fraca. Este pacote, chamado ANAGRAMA, também pretende ser utilizado na construção de compiladores e, desta maneira, são geradas as tabelas para os reconhecedores sintáticos. Uma introdução à teoria das linguagens formais é também apresentada.

ABSTRACT

The thesis presents a system for design and analysis of simple or weak precedence grammars and languages. This package, named ANAGRAMA, is also intended to be an aid to compilers construction. In this way, tables of sintatic recognizers are generated. Additionally, an introduction to formal languages theory is presented.

ÍNDICE

CAPÍTULO I

INTRODUÇÃO

O projeto de uma linguagem e do respectivo reconhecedor sintático é uma tarefa muitas vezes trabalhosa e demorada. Uma das principais razões é que a definição da aplicação a que se destina a linguagem, é inicialmente vaga e provisória, provocando assim, repetidas mudanças nas sintaxes propostas. Outro problema é que, uma vez caracterizada razoàvelmente a aplicação, a escolha da gramática exige um estudo cuidadoso, e cada proposta adotada requer inúmeras iterações de modo a se conseguir um analisador o mais eficiente e apropriado à aplicação considerada. Se este trabalho todo é feito fora da universidade pode haver ainda o transtorno adicional da falta de pessoas familiarizadas com este assunto. Isto se torna particularmente crítico em virtude de que o estudo formal da teoria da compilação é básicamente restrito às universidades, não havendo nenhum mecanismo já consolidado que possibilite um intercâmbio entre esta prática acadêmica e as necessidades surgidas em outros setores.

Já existe no Brasil uma certa demanda para a construção de compiladores em áreas não estritamente acadêmicas. Isto acontece em vários locais: em empresas que necessitam do compilador como componente em algum sistema maior e de uso interno; em empresas da nascente indústria nacional de minicomputadores que usam o compilador como produto acessório fornecido aos clientes; em empresas que comercializam programas, pacotes ou sistemas de computação.

O país atualmente deixa de ser mero usuário de caixas pretas importadas e já começa a dominar uma ainda incipiente tecnologia de computação. Neste momento os compiladores constituem uma área onde a autonomia pode ser rapidamente conquistada.

O presente trabalho tenta dar uma contribuição prática à disseminação da teoria dos compiladores e ser instrumento de ajuda aqueles que precisam projetar novas linguagens e seus respectivos reconhecedores. É oferecido um conjunto de programas na forma de três módulos, com a finalidade de analisar gramáticas livres do contexto segundo os conceitos de precedência. Estes três módulos são usados isoladamente, segundo critérios do próprio projetista/usuário e a entrada única para o sistema são as produções da gramática que se quer projetar. Como saída são obtidas tabelas do reconhecedor sintático, além de um conjunto de diagnósticos e mensagens, tudo na forma de relatórios a serem analisados pelo projetista/usuário. Este sistema que leva o nome de ANAGRAMA, tenta atingir dois objetivos:

- ser um pacote utilizável, tanto dentro como fora das universidades, no projeto de linguagens e compiladores;
- ser um instrumento didático para os cursos regulares de compiladores e linguagens formais, possibilitando um estudo e contato prático com as gramáticas/linguagens de precedência simples e fraca.

É fornecida também uma introdução teórica de linguagens formais, destinada aos não iniciados na matéria. Em seguida, vem a descrição global e básica do sistema, voltada mais para sua utilização prática. O restante do trabalho é dedica

do a descrever e documentar cada módulo e aí o nível de informação desce até a lógica de cada programa.

Foi empregada na descrição do sistema, uma técnica de documentação que é uma variante da técnica HIPO desenvolvida pela IBM (ver referência bibliográfica 10). No Apêndice I há uma rápida descrição dos diagramas e convenções utilizadas neste trabalho. Finalizando, há o Apêndice II que é um caso prático de uso do ANAGRAMA.

O pacote ANAGRAMA foi implementado em PL/I, em um IBM 370/158, em OS/VM. Ele é fruto básicamente do trabalho desenvolvido pelo autor no SERPRO nas áreas de sistemas de informações e linguagens, durante os anos de 1977 e 1978. O papel da COPPE também tem que ser destacado, principalmente pela atuação dos professores Guilherme Chagas Rodrigues e José Lucas Rangel, este último o orientador desta tese. É desejo do autor que este trabalho seja de utilidade, de alguma forma, em algum lugar, no desenvolvimento de linguagens e compiladores.

Cientistas esquartejam Puchkin e Baudelaire.
Exegetas desmontam a máquina da linguagem.
A poesia ri.

"A Poesia"

FERREIRA GULLAR

CAPÍTULO II

INTRODUÇÃO ÀS LINGUAGENS FORMAIS

II.1. Alfabeto

Definição : Um alfabeto é qualquer conjunto finito e não vazio de símbolos.

Nesta definição, símbolo é tomado como noção intuitiva básica, e neste texto sempre é usado como sinônimo e no lugar de elemento de um conjunto. O conjunto de algarismos V = { 1,2...,0 } é um exemplo de alfabeto.

II.2. Cadeia

Definição : Cadeia de Símbolos, ou simplesmente cadeia, é qualquer sequência finita de símbolos de um alfabeto V.

Uma cadeia é notada por uma letra grega minúscula. Dado o alfabeto V = { 1,2,...0 }, α = 12311 é um exemplo de cadeia.

Definição : O comprimento de uma cadeia é o número de símbolos que a constituem.

O comprimento da cadeia α é notado por $|\alpha|$. Se α = 12311, então $|\alpha| = 5$. É admitido aqui o conceito de cadeia com zero símbolos. Esta cadeia é chamada de cadeia vazia ou cadeia nula e é notada por ε. Desta forma $|\varepsilon| = 0$.

Definição : Dado um alfabeto V, o conjunto de cadeias de V é o conjunto de todas as cadeias de qualquer comprimento, possíveis de construir em V.

O conjunto de cadeias de V é notado por V^*. Se a cadeia nula é excluída, usa-se a notação V^+, significando $V^+ = V^* - \{ \varepsilon \}$.

Exemplo 1 : Seja V = { a,b,c } um alfabeto. Neste caso V* = { ε,a,b,c,aa,ab,ac... }.

II.3 - Produções

Definição. Seja um alfabeto V e o conjunto de cadeias V*. Uma produção é um par ordenado qualquer pertencente ao produto V* . V* .

O elemento da esquerda do par é chamado de raiz e o elemento da direita é chamado de argumento.

Definição. Um sistema de reescrita ou um conjunto de produções é uma relação binária P definida sobre V*, satisfazendo a condição que para todo $s \in V$ exista $\alpha, \beta, \gamma, \delta$ tal que $(\alpha\ s\ \beta, \gamma) \in P$ ou $(\gamma, \alpha\ s\ \beta) \in P$.
Se $(\alpha, \beta) \in P$ é dito então que α "pode ser reescrito como" β, que α "pode ser substituído por" β e isto é notado

$$\alpha ::= \beta .$$

Notar que a definição da relação P (ou ::=) assegura que todo símbolo do alfabeto figura em pelo menos uma produção, seja à esquerda (na raiz) ou à direita (no argumento).

P é um conjunto válido de regras mediante as quais é possível se reescrever ou substituir determinadas cadeias por outras.

II.4 - Derivações

Definição. Sejam $\alpha, \beta \in V^*$ e um conjunto de produções P em V*. Diz-se que α deriva diretamente β ou que há uma derivação direta de α para β, se existem $\gamma, \delta, \theta_1, \theta_2 \in V^*$ tal que $\alpha = \theta_1\gamma\theta_2$, $\beta = \theta_1\delta\theta_2$ e $\gamma ::= \delta$.

Se α deriva diretamente β, isto é notado por $\alpha \Longrightarrow \beta$.

Definição. Diz-se que α deriva β ou α produz β, se e somente se existe um conjunto de n cadeias, $n \geq 2$, $\theta_1, \theta_2, \theta_3, \ldots \theta_{n-1}, \theta_n$, tal que

$$\alpha = \theta_1,$$

$$\theta_1 \Rightarrow \theta_2,$$

$$\theta_2 \Rightarrow \theta_3,$$

$$\vdots$$

$$\theta_{n-1} \Rightarrow \theta_n,$$

$$\theta_n = \beta$$

Portanto, α deriva β se e somente se há uma sequência de derivações diretas, tal que α é elemento da esquerda da primeira derivação direta e β é o elemento da direita da última derivação direta.

Se α deriva β, isto é notado por $\alpha \overset{+}{\Rightarrow} \beta$.
Se é possível $\alpha=\beta$, então α deriva β é notado $\alpha \overset{*}{\Rightarrow} \beta$.

As condições "derivar diretamente" e "derivar" definem duas novas relações binárias em V*. Se α deriva diretamente β, isto acarreta que α deriva β. O inverso não é verdadeiro. Assim a relação "derivar diretamente" é um subconjunto da relação "derivar".

Exemplo 2. Dado o alfabeto V = { a, b, c } e as produções abc ::= c (1), ac ::= ε (2), b ::= a (3), aplicar as sequências de produções 1,2,3,2 e 3,3,2,2 à cadeia aabcbc.

Para a primeira sequência obtem-se :

aabcbc => acbc => bc => ac => ε ; para a segunda obtem-se: aabcbc => aaacbc => aaacac => aaac => aa

O tipo de aplicação aqui estudada não precisa que uma cadeia seja reduzida à cadeia nula ou a uma cadeia de comprimento menor. Neste exemplo isto aconteceu devido a que as produções, da maneira como estão definidas, são absolutamente gerais, permitindo qualquer tipo de substituições e reescritas. É de interesse, pois, que sejam impostas algumas restrições às produções para que as derivações sejam mais sistemáticas, podendo ser melhor estudadas e se prestando mais ao estudo dos compiladores mais usados atualmente.

II.5 - Produções Livres do Contexto

Definição. Um conjunto de produções P é dito livre do contexto se para cada produção $\alpha ::= \beta$ tem-se

$$| \alpha | = 1 .$$

Definição. Um conjunto de produções P livre do contexto é restrito se $\alpha \neq \beta$ e $\beta \neq \varepsilon$.

Nesse estudo todo conjunto de produções P é livre do contexto e restrito, a não ser que se declare explicitamente o contrário.

Foram agora introduzidas severas restrições ao conjunto de produções P. A raiz da produção passa a ser uma cadeia de apenas um símbolo e o argumento será uma cadeia não nula e diferente da raiz. Pode se deduzir de imediato duas consequências importantes destas restrições: um P não produz cadeias nulas nem diminui o comprimento de uma cadeia ao longo de uma derivação. E por que produções com estas restrições são chamadas de "livres do contexto" ? Porque o símbolo de que se compõe a raiz de cada produção pode ser substituído pelo respectivo argumento em

qualquer cadeia em que ele estiver presente sem levar em conta o contexto em que ela possa ocorrer. Uma produção da forma $\theta_1\alpha\theta_2 ::= \theta_1\beta\theta_2$, por exemplo, especifica que α pode ser substituído por β se e somente se α ocorrer no contexto $\theta_1 \ldots \theta_2$. Produções com esta característica são por isto mesmo chamadas de sensíveis ao contexto, mas não serão estudadas neste texto.

II.6 - Conjunto Terminais e Não-Terminais

Definição. Dado um alfabeto V chamam-se respectivamente conjuntos terminais e não-terminais a dois Conjuntos T e N, tais que

$$1 - \quad T \cup N = V ,$$

$$2 - \quad T \cap N = \phi ,$$

$$3 - \quad T \neq \phi \text{ e } N \neq \phi ,$$

onde ϕ representa o conjunto vazio .

Os conjuntos terminais e não-terminais são nada mais que uma forma alternativa de apresentar o alfabeto que fica assim particionado em dois subconjuntos. A utilidade e o significado deste procedimento ficarão claros logo a seguir.

Será normal referir-se a T* (conjunto das cadeias formadas de terminais), a N* (conjunto das cadeias formadas por não-terminais) e a (N ∪ T)* (conjunto de cadeias quaisquer).

Neste texto será adotada a seguinte notação

1 - A, B, C, D ... para não-terminais ;

2 - a, b, c, d ... para terminais ;

3 - s, t, u, v ... para qualquer símbolo ;

4 - x, y, z, w ... para cadeias de terminais.

II.7 - Gramática

Definição. Uma gramática livre do contexto G é uma quadrupla G = (N, T, P, S) , onde

1 - N é o conjunto dos não-terminais ;

2 - T é o conjunto dos terminais ;

3 - P é um conjunto de produções definido em $N^* \cdot (N \cup T)^*$.

4 - $S \in N$ é chamado o símbolo inicial de G .

O símbolo inicial, em princípio, é qualquer não-terminal . No entanto, para certos tipos de aplicação, é importante que o símbolo inicial seja um não-terminal que preencha determinadas condições. Será adotada então a seguinte definição.

Definição. Seja um alfabeto V e um conjunto de produções P definido de tal maneira que exista um e somente um não-terminal S, preenchendo as seguintes condições :

1. não existe uma produção da forma S ::= x, (lembrar que $x \in T^*$),

2. não existe uma produção da forma A ::= αSβ,

Esse não-terminal é chamado de símbolo inicial .

O símbolo inicial geralmente é representado pela letra S neste texto, a não ser que explicitamente se use outra representação.

Exemplo 3. Dado o alfabeto V = { a,b,c,d,e,f,g } e P composto de e ::= abc , e ::= bfg , f ::= c e g ::= d , determinar os conjuntos não-terminal e terminal e o símbolo inicial.

Tomando a definição de terminal, não-terminal e gramática os

conjuntos pedidos são os seguintes :

$$T = \{ a,b,c,d \} \quad e \quad N = \{ e,f,g \} .$$

Não há símbolo inicial, pois o único não-terminal que não figura em qualquer argumento (o não-terminal e), figura em uma produção com o argumento composto só de terminais (e ::= abc) .

II.8 - Linguagens

Definição. Dado um alfabeto V, uma linguagem sobre V é qualquer subconjunto de V* .

Dada uma gramática G, um certo tipo de linguagem é de particular interesse e aplicação.

Definição. A linguagem terminal gerada por uma gramática G é o conjunto

$$L = \{ x \mid S \overset{+}{\Rightarrow} x \} ,$$

onde S é o símbolo incial de G .

Notar que uma derivação sempre pode continuar enquanto nas cadeias sucessivamente geradas ainda figurar algum não-terminal. O fim irremediável de uma derivação é, pois, uma cadeia só de terminais. Daí a propriedade essencial das linguagens terminais: ser um conjunto de cadeias finais das derivações que se iniciam no símbolo inicial.

Esse estudo se interessa apenas pelas linguagens terminais, que passarão a ser chamadas daqui por diante simplesmente de linguagens.

É importante notar que uma determinada gramática gera uma única linguagem, mas o inverso não é verdadeiro. Dado um determinado conjunto de cadeias considerado como uma linguagem, é possível

definir-se mais de uma gramática geradora desta linguagem. Dependendo então da aplicação, é escolhida a gramática mais adequada. A linguagem assume, assim, os atributos da gramática. É importante notar que existem linguagens às quais não corresponde nenhuma gramática livre do contexto. As linguagens geradoras por gramáticas livres do contexto são chamadas linguagens livres do contexto e serão as únicas de interesse a este estudo.

Exemplo 4. Dada G = (N,T,P,S), onde N = { S,A }, T = { 1 } e P composto de S ::= , A ::= A1 , A ::= 1 , descrever a linguagem gerada por G .

A linguagem é o conjunto de cadeias compostas de 1's .

Exemplo 5. Dada G = ({ 1,2,3 } , { 4 } { 1 ::= 2, 2 ::= 32 , 3 ::= } , 1) descrever a linguagem gerada por G .

A linguagem é o conjunto vazio. Notar que é impossível gerar qualquer cadeia de terminais a partir de 1 .

II.9 - Reconhecimento Sintático

Definição. Uma forma sentencial de uma gramaática G é uma cadeia derivável do símbolo inicial, isto é, α é uma forma sentencial se existe a derivação

$$S \overset{*}{\Rightarrow} \alpha \ .$$

Definição. Uma sentença é uma forma sentencial α, tal que

$$\alpha \in T^* \ .$$

A linguagem L é, portanto, o conjunto de sentenças de uma gramática G .

O estudo até aqui apresentado tenta resolver o seguinte problema:

- como encontrar uma maneira sistemática de construir um determinado conjunto de cadeias de símbolos de um dado alfabeto ?

A "maneira sistemática de construir" é a gramática e o "determinado conjunto de cadeias" é a linguagem. Mas, imediatamente surge outra questão:

- dada uma gramática, como reconhecer se uma cadeia pertence à linguagem gerada por esta gramática ?

O restante desse estudo, praticamente, é devotado a resolver esta questão.

Dada uma cadeia α e uma gramática G, surge um critério de se saber se α é uma forma sentencial de G, ou se $\alpha \in T^*$, saber se α é uma sentença de G :

- α é uma forma sentencial de G se for possível construir uma derivação partindo do símbolo inicial e terminado em α .

É importante notar que o conceito de símbolo inicial fornece um critério definitivo para a verificação se uma sentença pertence ou não a uma linguagem, já que a derivação desta sentença se inicia nele. Além de fornecer o critério de ação para o reconhecimento sintático, o símbolo inicial também facilita a implementação de técnicas para este reconhecimento.

Definição. O reconhecimento sintático de uma cadeia α segundo uma gramática G é a verificação da existência ou não existência, de uma derivação $S \overset{*}{\Rightarrow} \alpha$, onde S é o símbolo inicial de G.

Quando a gramática G está implícita, diz-se simplesmente "reconhecimento sintático de α".

Quando α é tal que $\alpha \in T^*$, o reconhecimento sintático determinará se α pertence ou não à linguagem em questão. Uma cadeia de símbolos terminais poderá ser chamada de sentença, mesmo se o reconhecimento sintático ainda não foi efetuado. Portanto, será habitual e aceitável se dizer "reconhecimento sintático da sentença". Por outro lado "reconhecimento sintático" será geralmente abreviado para "reconhecimento", simplesmente.

II.10 - Frase

Definição. Seja uma gramática G e $\alpha = \theta_1 \beta \theta_2$ uma forma sentencial. A cadeia β é uma frase α para um não-terminal A se existem as derivações :

$$S \overset{*}{\Rightarrow} \theta_1 A \theta_2 \quad \text{e} \quad A \overset{*}{\Rightarrow} \beta \ .$$

Se A ::= β, então β é uma frase simples.
O fato de $A \overset{*}{\Rightarrow} \beta$ não implica que β seja uma frase de $\alpha = \theta_1 \beta \theta_2$ para o não-terminal A. É necessário também que $S \Rightarrow \theta_1 A \theta_2$.

Uma forma sentencial α poderá ter várias frases para vários não-terminais.

Definição. O peão de uma forma sentencial é a frase simples que se encontra mais à esquerda.

Exemplo 6. Seja G = (N,T,P,S), onde N = {S,M,D},
T = {1,2,3}, e P composto de S ::= M, M ::= MD, M ::= D,
D ::= 1, D ::= 2 e D ::= 3.

Será M uma frase da forma sentencial M1 para o não-terminal S ?
E qual o peão da forma sentencial D2D ?

Realmente existe a derivação S $\overset{*}{\Rightarrow}$ M, mas não existe a derivação S $\overset{*}{\Rightarrow}$ S1. Logo M não é a frase de M1 para S.

O terminal 1, no entanto, é frase de M1 para o não-terminal D, pois S $\overset{*}{\Rightarrow}$ MD e D $\overset{*}{\Rightarrow}$ 1. Aliás, 1 é frase simples, pois D ::= 1.

A forma sentencial D2D possue as seguintes frases:

(1) 2 é uma frase para o não-terminal D, pois S $\overset{*}{\Rightarrow}$ DDD e D $\overset{*}{\Rightarrow}$ 2;

(2) D é uma frase para o não-terminal M, pois S $\overset{*}{\Rightarrow}$ MDDD e M $\overset{*}{\Rightarrow}$ D;

As duas frases são também frases simples, pois D ::= 2 (caso 1) e M ::= D (caso 2). A frase simples mais a esquerda é 2. Logo o peão é 2.

II.11 - Derivação canônica

Definição. Dada uma sentença α, a derivação canônica de α é sequência de derivações diretas S => θ_1 ... α, S o símbolo inicial, tal que a aplicação das produções às formas sentenciais seja sempre feita no não-terminal mais a esquerda.

O reconhecimento sintático de uma cadeia α através da derivação S $\overset{*}{\Rightarrow}$ α (S, símbolo inicial) pode ser obtido de duas formas:

(1) partindo de S para reconstruir α;

(2) partindo de α para reconstituir S

Se a derivação que se tenta construir é a canônica surge um método sistemático de conseguí-la. Para a primeira das duas formas citadas acima, o método consiste em se ir gerando peões a partir de sucessivos não-terminais e com estes peões ir recons-

truindo todas formas sentenciais até chegar a α.

No segundo caso, toma-se o caminho inverso, isto é começando em α, ir reduzindo sucessivamente os peões a não-terminais, até chegar a S.

Os métodos realmente utilizados nos compiladores para o reconhecimento sintático se diferenciam pouco destas duas maneiras básicas. Daí a notável importância dos conceitos de frase e peão.

Exemplo 7. Seja a gramática do exemplo 6. Fazer o reconhecimento da sentença 12.

É possível a seguinte derivação:

S => M => MD => M2 => D2 => 12 .

Realmente, a sentença 12 pertence a linguagem gerada pela gramática. Esta derivação não é canônica. A derivação canônica é a seguinte:

S => M => MD => DD => 1D => 12 .

II.12 - A Notação BNF

Foi visto que a linguagem gerada por uma gramática é um conjunto de cadeias de terminais. Existe pois uma diferença prática entre os conjuntos terminal e não-terminal. O símbolo não-terminal é, essencialmente, uma variável que na formação das sentenças da linguagem vai sendo substituída por outros símbolos ao longo da derivação. É por isto que um não-terminal é, por vezes, chamado de "entidade sintática","variável sintática" ou "símbolo variável". O terminal, como se sabe, não pode ser substituído nunca.

Há também uma necessidade prática de não se confundir os símbo-

los da própria 'linguagem' natural (português, inglês, etc) utilizada nos textos com os símbolos da linguagem formal sendo estudada.

Existe uma notação para descrever uma gramática e os processos dela decorrentes, que explicita a diferença citada entre terminal e não-terminal, além de atenuar bastante a confusão entre os símbolos da gramática estudada e dos textos. Esta notação é baseada na seguinte convenção gráfica:

1. os símbolos não-terminais serão referidos por um ou vários nomes esclarecedores de suas próprias funções na gramática e estes nomes serão escritos entre os símbolos " < " e " > " ; na gramática do exemplo 6 o não-terminal D poderia ser referido como <dígito>;

2. quando várias produções possuem a mesma raiz elas poderão ser abreviadas de tal modo que os diferentes argumentos figurem à direita do símbolo ::= separados pelo símbolo "|"; na gramática do exemplo 6 as produções D ::= 1, D ::= 2, D ::= 3, poderiam ser referidas como

D ::= 1 |2| 3 ou ainda

<dígito> ::= 1 |2| 3 .

A notação de uma gramática, portanto, utilizará os seguintes símbolos descritivos:

SÍMBOLO DESCRITIVO	SIGNIFICAÇÃO	USO
::=	é definido como	separa o não-terminal de sua definição
|	ou	separa definições alternativas de um mesmo não-terminal.
<xyz>	"xyz"	indica que xyz devem ser vistos como nome do texto

Esta notação será chamada de notação BNF, do inglês BACKUS-NAUR FORM ou BACKUS NORMAL FORM e foi desenvolvida por John Backus para descrição do ALGOL 60.

Exemplo 8. Utilizando a notação BNF, descrever a gramática do exemplo 6.

Poder-se-ia ter G = (N,T,P, <número>), onde

N = { <número> , <raiz> , <dígito> },

T = { 1,2,2 } e P é composto de

<número> ::= <raiz> ,

<raiz> ::= <raiz> <dígito> | <dígito> ,

<dígito> ::= 1|2|3 .

II.13 - Relações de Precedência

O reconhecimento sintático de interesse a este texto é aquele que procura descobrir se uma determinada cadeia de terminais pertence ou não à linguagem gerada por uma gramática. Além disto, o reconhecimento é feito sempre segundo uma derivação canônica. Como consequência, e segundo o que foi visto em II.11, o problema do reconhecimento se resume, a cada instante, em des

cobrir o peão de uma forma sentencial. Esta descoberta, pelo que foi mostrado, é possível ser feita por tentativas, sendo que para uma gramática com muitas produções este processo é muito demorado, o que o torna sem valor prático.

A grande dificuldade do processo de tentativas é que, ao se ir percorrendo a forma sentencial da esquerda para a direita, não se sabe se uma redução possível de ser feita é uma redução correta. Dito de outra forma, não se sabe se uma cadeia que pode ser reduzida é realmente o peão.

Como acelerar a procura do peão? Uma idéia é descobrir alguma relação entre os símbolos da gramática, de tal forma que a cada etapa do reconhecimento os limites do peão possam ser localizados mediante consulta a esta relação. Mais precisamente, a relação entre o último símbolo lido e o próximo deve indicar, por exemplo, se o fim do peão já foi atingido ou não. Se a relação indicar fim do peão a redução deve ser feita; se não, o próximo símbolo deve ser lido e empilhado.

Que relação é esta? Ela é chamada de MAIOR e é usada no reconhecimento para localizar o último símbolo à direita do peão de uma forma sentencial.

Existem ainda duas outras relações, MENOR e IGUAL, que possibilitam determinar o início do peão e qual a redução exata a ser feita.

Seja $\alpha = \beta\theta x$ uma forma sentencial de uma derivação canônica de uma gramática G e seja θ o peão. Lembrando que $x \in T^*$ tem-se:

1 - a relação MAIOR existe entre o último símbolo de θ e o primeiro símbolo de x ;

2 - a relação MENOR existe entre o último símbolo de β e o primeiro símbolo de θ ;

3 - a relação IGUAL existe entre dois símbolos vizinhos contido em θ .

Estas três relações são chamadas de relações de precedência pois indicam, a cada instante, que símbolos são reduzidos primeiro que outros.

A localização do peão no reconhecimento sintático é feito percorrendo-se a forma sentencial da esquerda para a direita até que a relação MAIOR seja encontrada. Em seguida retorna-se a procura até encontrar a relação MENOR. O peão é a cadeia entre MENOR e MAIOR. Se a gramática é tal que não existam duas produções com o mesmo argumento então a redução é única e portanto determinada. Neste caso o processo pode ser repetido até que se atinja o símbolo inicial ou até que não se possa fazer mais nenhuma redução.

II.14 - Relações PRIMEIRO, ÚLTIMO e IGUAL

Definição. Seja uma gramática $G = (N,T,P,S)$ e as três seguintes relações entre seus símbolos:

$$1 - \text{PRIMEIRO} = \{ (A,v) \mid (A ::= v\alpha) \in P \} ,$$

$$2 - \text{ÚLTIMO} = \{ (A,v) \mid (A ::= \alpha v) \in P \} ,$$

$$3 - \text{IGUAL} = \{ (u,v) \mid (A ::= \alpha u v \beta) \in P \} .$$

É evidente que: (a) PRIMEIRO existe entre cada raiz e o primeiro símbolo do respectivo argumento; (b) ÚLTIMO existe entre cada raiz e o último símbolo do respectivo argumento; (c) IGUAL existe entre dois símbolos vizinhos em um argumento.

Qual o número máximo de pares em cada uma destas três relações?
Dada uma gramática G, sejam as seguintes quantidades :

a : soma dos tamanho dos argumentos das produções;

p : número de produções;

max | R | : número máximo de pares na relação.

Então: max | IGUAL | = a - p

max | PRIMEIRO | = max | ÚLTIMO | = p .

A partir das definições não é difícil provar estas igualdades.

II.15 - Relações MAIOR E MENOR

Definição: Dada uma gramática G = (N,T,P,S), sejam definidas as relações MENOR e MAIOR

1 - MENOR = IGUAL x PRIMEIRO^{+}

2 - MAIOR = (ÚLTIMO^{-1})$^{+}$ x IGUAL x PRIMEIRO*,
definida em (N$\cup$T).T

Lembrar que se R é uma relação, R^{+} é o fechamento transitivo de R, isto é, a R^{+}b se existe uma sequência de elementos $c_1, c_2 \ldots, c_k$, $k \geq 1$, tal que $a = c_1$, c_1 R c_2, c_2 R c_3 ... c_{k-1} R c_k, $b = c_k$. O fechamento transitivo e reflexivo de R, notado por R^{*}, é definido por a R^{*}b se e somente a = b ou a R^{+}b. Se A PRIMEIRO^{+} u então existe uma derivação A $\overset{+}{\Rightarrow} u\alpha$
Se A ÚLTIMO^{+} u então existe uma derivação A $\overset{+}{\Rightarrow} \alpha u$
Notar que o operador x indica o produto de relações e lembrar, portanto, que este produto é efetuado da esquerda para a direita, de modo que a R_1 x R_2 b se e somente se existe algum c tal que a R_1 c e c R_2 b. Se a R b, então b R^{-1}a é a relação inversa de R.

A determinação de MENOR e MAIOR poderá, obviamente, ser feita determinando-se IGUAL, PRIMEIRO e ÚLTIMO diretamente da gramática e fazendo-se as composições como indica a definição. Para isto pode-se empregar o método de Warshall para se efetuar o fechamento transitivo e um algoritmo geral de multiplicação de matrizes booleanas para se efetuar o produto das relações. Estes métodos utilizam representar a relação como uma matriz booleana e são descritas em GRIES[6].

As relações podem também ser representadas por um grafo e em HUNT[7] é descrito um métodos que implementa operações em relações (fechamento transitivo, inversão, produto e união) como correspondentes operações em grafos.

Ainda segundo HUNT[7], dada uma gramática G, onde a é a soma dos tamanho dos argumentos e p é o número de produções,

então

$$\max \mid MAIOR \mid = 4 \ (a + p) \ .$$

De HUNT[7] pode também ser deduzido que

$$\max \mid MENOR \mid = 2 \ (a + p) \ .$$

A relação MAIOR como definida em HUNT[7] contém a relação MAIOR como definida aqui, pois lá são admitidos pares (u,v) tal que $v \in N$, e aqui não. Como consequência, na realidade $\max \mid MAIOR \mid \leq 4 \ (a + p)$. Apesar disto, este valor máximo será utilizado neste texto.

II.16 - Gramáticas de Precedência Simples

Definição: Uma gramática G = (N,T,P,S) é dita ser de precedência se existir, no máximo, uma das três relações IGUAL, MENOR e MAIOR entre dois símbolos quaisquer do alfabeto.

Definição: Uma gramática G = (N,T,P,s) de precedência é dita ser de precedência simples se não existir duas produções com o mesmo argumento.

No reconhecimento, é conveniente acrescentar aos símbolos da gramática um símbolo delimitador de cadeia para indicar o início e o fim da sentença.

Este delimitador, aqui notado por $, é tal que

1 - $ MENOR u, sempre que $S \overset{*}{\Rightarrow} u\alpha$

2 - v MAIOR $, sempre que $S \overset{*}{\Rightarrow} \alpha v$.

Dito de outra forma, MENOR relaciona $ com os símbolos que podem iniciar uma forma sentencial válida e MAIOR relaciona os símbolos que podem terminá-la com $. Este símbolo delimitador é considerado um terminal e de agora por diante é convencionado que o conjunto T sempre o inclui.

II.17 - Gramáticas de Precedência Fraca

Muitas gramáticas apresentam conflitos de precedência entre seus símbolos apenas do tipo MENOR - IGUAL. Como se sabe, no reconhecimento estas duas relações indicam que o fim do peão ainda não foi atingido, devendo o símbolo lido ser empilhado. Se as duas relações são unidas, poder-se-á ter uma nova relação que indica simplesmente que o símbolo lido deve ser empilhado. A relação MAIOR continuará a ser usada para indicar que a redução deve ser feita e o peão é exatamente encontrado procurando-se uma produção cujo argumento coincida com os últimos símbolos empilhados.

Definição: Seja G = (N,T,P,S) satisfazendo as seguintes condições :

1 - MAIOR $\cap$ (MENOR $\cup$ IGUAL) = ϕ (ϕ o conjunto vazio),

2 - Se (A ::= $\alpha u \beta$) $\in$ P e (y ::= β) $\in$ P, então não existe relação de precedência entre u e Y

G então é dita ser de precedência fraca.

Porque a condição 2 nesta definição? Para que nunca haja ambigüidade na hora de se efetuar a redução. Se na hora de reduzir, existir $\alpha u \beta$ no topo da pilha, o que deverá ser feito? Reduzir β para Y ou $\alpha u \beta$ para A ? Se a condição 2 é satisfeita, a redução sempre deverá ser feita para a produção de maior argumento.

Em resumo, o reconhecimento que usa gramáticas de precedência fraca apenas verifica as relações existentes entre o último símbolo empilhado e o próximo símbolo a ser lido. Como o próximo símbolo sempre é um terminal, pois o reconhecimento usa uma derivação canônica a partir de uma sentença, nas gramáticas de precedências fraca basta que as relações IGUAL e MENOR sejam definidas em (N$\cup$T).T .

II.18 - Matriz de Precedência

Os símbolos de uma gramática G podem ser associados a uma sequência de inteiros $1 \leq i \leq n$, onde n é o número de símbolos de G, tal que cada símbolo s_i possa ser referido pelo inteiro i . Desta maneira, as relações de precedência de G podem ser representadas por uma matriz M, chamadas matriz de precedência, tal que a entrada M_{ij} fornece a relação entre s_i e s_j.

As entradas de M podem assumir diversos valores, três deles representando MENOR, IGUAL e MAIOR, outro indicando a não existência de precedência, e outros que sejam necessários para re-

presentar conflitos de precedência.

No caso de gramática de precedência simples, é comum M ser uma matriz numérica. Neste trabalho é convencionado que

$M_{ij} = 1$, se s_i MENOR s_j ,

$M_{ij} = 2$, se s_i IGUAL s_j ,

$M_{ij} = 3$, se s_i MAIOR s_j e

$M_{ij} = 0$, para as outras entradas.

Quando $M_{ij} = 0$, isto indica que os símbolos s_i e s_j não podem figurar vizinhos numa forma sentencial gerada pela gramática estudada.

Se a gramática é de precedência simples, M será uma matriz quadrada, onde o número de linhas e colunas é o número de símbolos de G. Se a precedência é fraca, IGUAL e o conflito IGUAL/MENOR são tratados com MENOR o número de linhas é o número de símbolos e o número de colunas é o número de terminais.

As formas de armazenamento e compactação da matriz de precedência são descritas no Capítulo VI.

Penetra surdamente no reino das palavras.
Lá estão os poemas que esperam ser escritos.
Estão paralisados, mas não há desespero,
há calma e frescura na superfície intata.
Ei-los sós e unidos, em estado de dicionário.

"Procura da Poesia"

CARLOS DRUMOND DE ANDRADE

CAPÍTULO III

ANAGRAMA - UM ANALISADOR DE GRAMÁTICAS DE PRECEDÊNCIA

O uso de compiladores puramente orientados para a sintaxe tem o inconviniente de limitar e privar a linguagem projetada de facilidades úteis e muitas vezes indispensáveis. Por outro lado, a teoria de linguagens formais põe à disposição do projetista de compiladores um arsenal de algoritmos e técnicas, entre as quais se encontram as aqui apresentada, que facilitam o projeto e desenvolvimento de novas linguagens.

III.1 - Definição Geral do Sistema

O Sistema ANAGRAMA é um analisador de gramáticas livres de contexto tendo em vista os conceitos de precedência. A partir das produções da gramática, são fornecidos relatórios de diagnósticos e orientação objetivando o projeto final de uma linguagem de precedência simples ou fraca. Deste modo, o usuário tem condições de ir modificando as produções da gramática através do uso iterativo do sistema, tendo em vista os objetivos pretendidos. Quando a gramática está finalmente pronta, o usuário tem a seu dispor as produções codificadas e a tabela do reconhecedor sintático escolhida dentre várias alternativas oferecidas pelo sistema ANAGRAMA.

III.2 - Descrição da Entrada

A entrada do ANAGRAMA é o conjunto de produções da gramática a ser analisada. Estas produções devem estar num arquivo em disco que pode ser gerado a partir de um arquivo em

A N A G R A M A - ANALISADOR DE GRAMATICAS DE PRECEDENCIA (DEFINICAO GERAL DO SISTEMA)

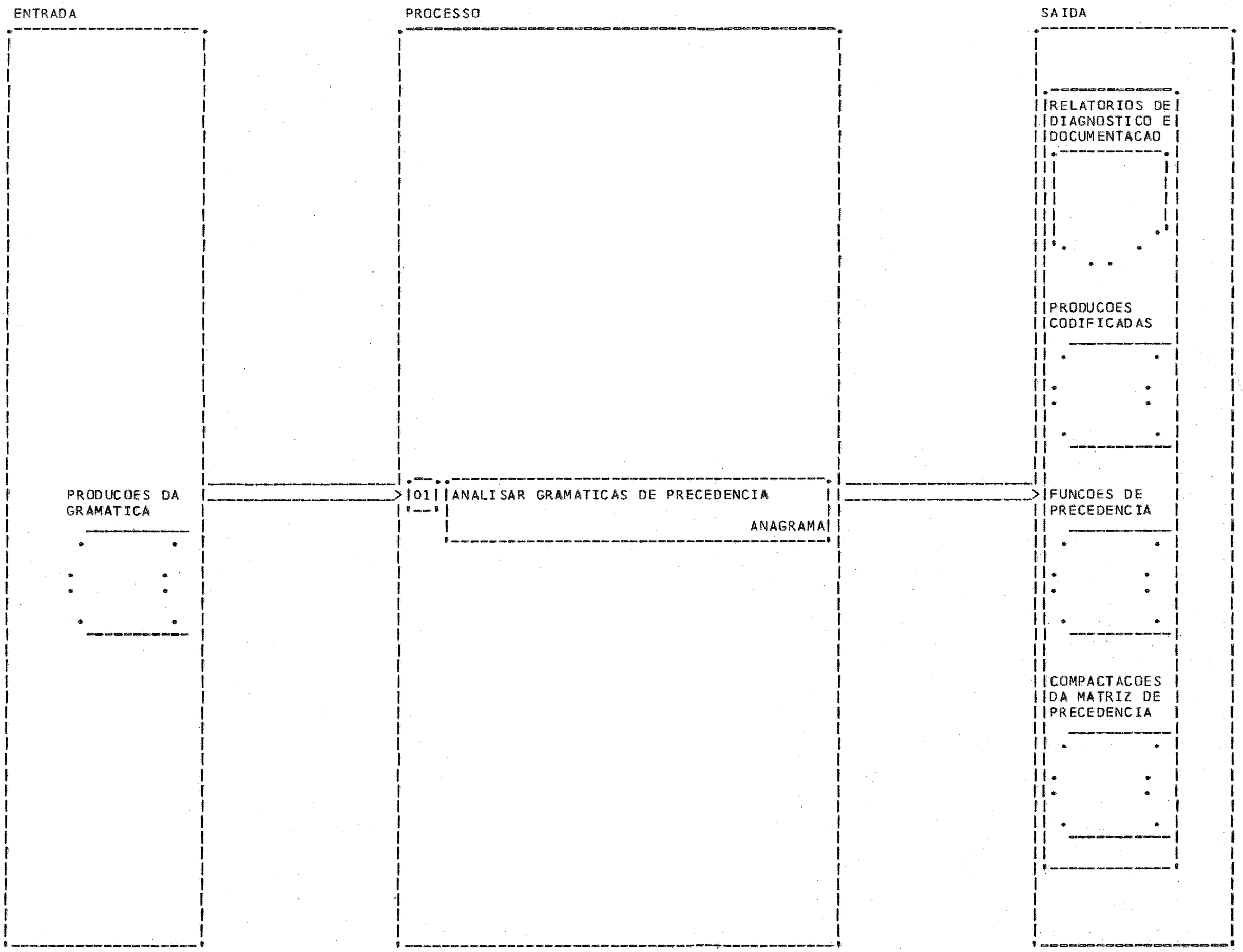


FIGURA III.1

cartões usando-se um procedimento apropriado (ver Fig. III.1).

III.3 - Descrição das Saídas

Vendo o sistema como um todo, são consideradas saídas os relatórios de análise e documentação e as tabelas a serem usadas pelo reconhecedor sintático. Deste modo, as saídas do ANAGRAMA podem ser básicamente as seguintes (ver Fig. III.1):

(1) relatórios de diagnósticos e documentação;

(2) produções codificadas;

(3) funções lineares de precedência;

(4) compactações alternativas da matriz de precedência.

III.3.1 - Relatórios de Diagnósticos e Documentação

O sistema emite três relatórios, chamados formalmente de Relatórios 01, 02 e 03. O primeiro deles serve de documentação para a gramática. Nele as produções de entrada são listadas e editadas segundo a notação BNF. Os conjuntos terminais e não-terminais, as relações de precedência entre os símbolos, diagnósticos e mensagens de erro são também listados. O segundo relatório imprime os valores das funções lineares de precedência, se elas existem. Finalmente, o Relatório 03 fornece os resultados das diversas compactações efetuadas na matriz de precedência.

III.3.2 - Produções Codificadas

Cada símbolo da gramática é codificado como um número inteiro e o conjunto das produções já adotando esta codificação ficam prontas para serem usadas pelo reconhecedor

sintático. As produções assim codificadas são gravadas em um arquivo em disco.

III.3.3 - Funções Lineares de Precedência

É um arquivo em disco contendo os valores das funções lineares de precedência da gramática, pronto para ser utilizado pelo reconhecedor sintático. Este arquivo é composto de duas partes: os controles da gramática e os valores das funções. Os controles são dados característicos da gramática e se compõem do número de terminais, número de não-terminais, número de produções, tamanho acumulado dos argumentos, símbolo inicial e tipo de precedência.

III.3.4 - Compactações da Matriz de Precedência

Básicamente são efetuados dois procedimentos de compactação na matriz de precedência. O primeiro procedimento tenta reduzir as linhas e colunas iguais de uma matriz, gerando uma segunda matriz dita reduzida. A idéia do segundo procedimento é representar compactadamente grupos de elementos iguais dentro de uma mesma linha da matriz. Cada grupo destes é então representado por um par, composto da ordem da coluna do último elemento do grupo e do próprio elemento do grupo. A partir destes procedimentos básicos são fornecidos ao projetista/usuário três tipos de compactação da matriz de precedência:

(1) redução da matriz original, resultante da aplicação do primeiro procedimento à matriz de precedência; são gerados a matriz reduzida e dois veto-

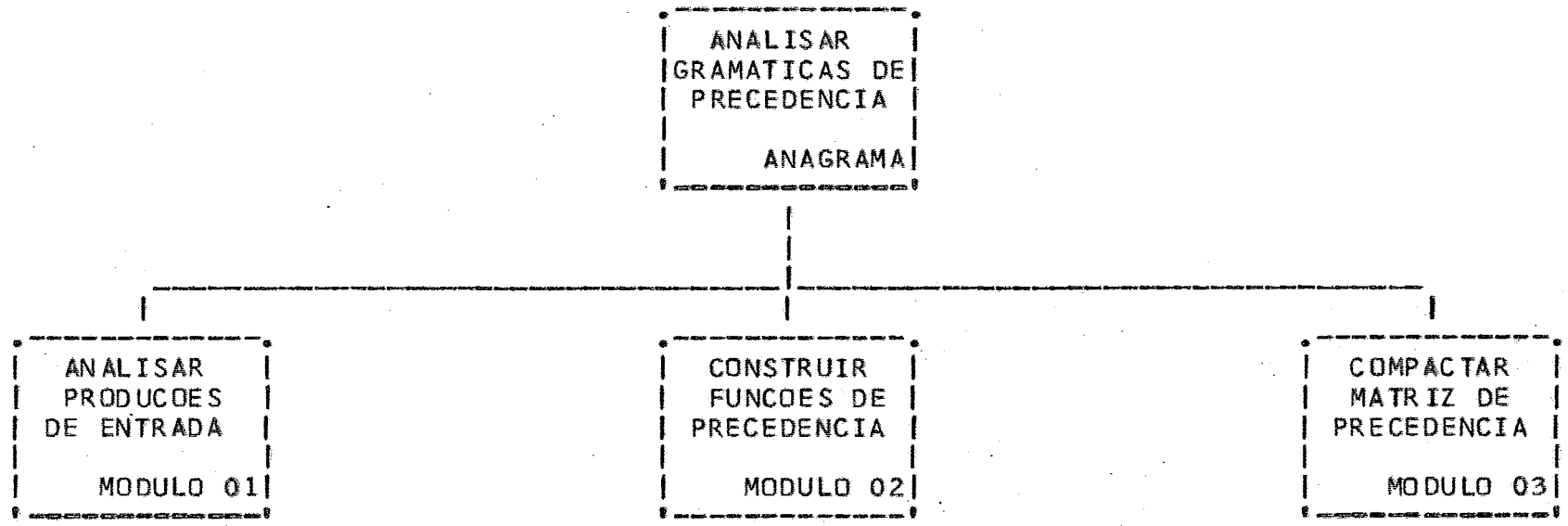

ANALISAR GRAMATICAS DE PRECEDENCIA
ANAGRAMA
ANALISAR PRODUCOES DE ENTRADA
MODULO 01
CONSTRUIR FUNCOES DE PRECEDENCIA
MODULO 02
COMPACTAR MATRIZ DE PRECEDENCIA
MODULO 03

FIGURA III.2

res redutores de linha e coluna; estes vetores associam a linha ou coluna da matriz original com a correspondente linha ou coluna da matriz reduzida;

(2) compactação da matriz original, resultante da aplicação do segundo procedimento à matriz original; são gerados uma sequência de pares representando os grupos de elementos iguais das linhas da matriz e um vetor cujos valores apontam para o primeiro par de cada linha;

(3) compactação da matriz reduzida, resultante da aplicação do segundo procedimento à matriz reduzida; são gerados a sequência de pares das linha da reduzida, um vetor cujos valores apontam para o primeiro par correspondente a cada linha e um vetor redutor de colunas.

A descrição detalhada destes três tipos de compactação bem como das respectivas formas de acesso fazem parte do Capítulo VI deste trabalho. A própria matriz de precedência original, não compactada, é também oferecida pelo ANAGRAMA, para os casos em que a compactação não tenha sido vantajosa. Cada um dos três tipos de compactação e a matriz original, constitue um arquivo em disco.

III.4 - Módulos do Sistema

O sistema é composto de três módulos (ver Fig.III.2), com as seguintes funções:

- Módulo 01 - Analisar Produções de Entrada
- Módulo 02 - Construir as Funções de Precedência
- Módulo 03 - Compactar a Matriz de Precedência

Cada um destes módulos constitue um pacote isolado utilizável independentemente pelo projetista segundo seu próprio citério de escolha. O Módulo 01 é o primeiro a ser executado e a partir dos resultados emitidos as produções da gramática podem ser modificadas. Novamente o módulo pode ser executado e o processo se repete até que se consiga uma gramática de precedência simples ou precedência fraca. Em seguida, é possivel que o projetista queira usar funções de precedência no seu reconhecedor. Neste caso, o Módulo 02 é o seguinte a ser executado. Dependendo do resultado, o Módulo 03 possibilita a escolha do melhor tipo de compactação. De qualquer maneira, após o Módulo 01, o projetista pode escolher livremente se usará o Módulo 02 ou Módulo 03 (ver Figs. III. 3, 4, e 5).

III.5 - Formato da Entrada

A entrada do ANAGRAMA é um arquivo contendo as produções da gramática a ser analisada. Este arquivo é gerado de acordo com as seguintes regras:

(1) codificação das produções:

a) cada produção é completamente escrita e contida em um registro de 80 caracteres, não havendo pois registros de continuação;

b) a gramática deve conter no máximo 255 símbolos;

A N A G R A M A - ANALISADOR DE GRAMATICAS DE PRECEDENCIA

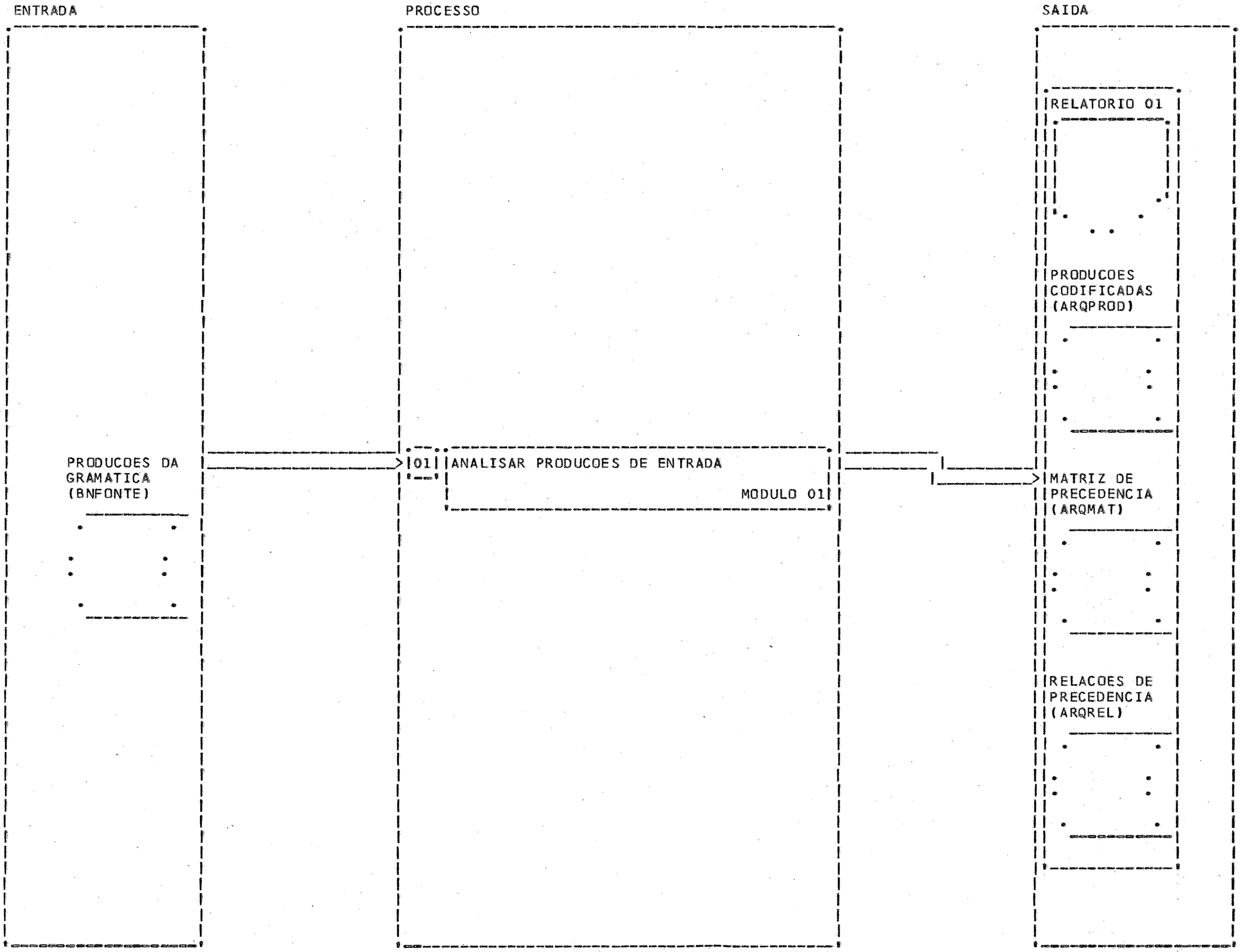


FIGURA III.3

A N A G R A M A - ANALISADOR DE GRAMATICAS DE PRECEDENCIA (CONT.)

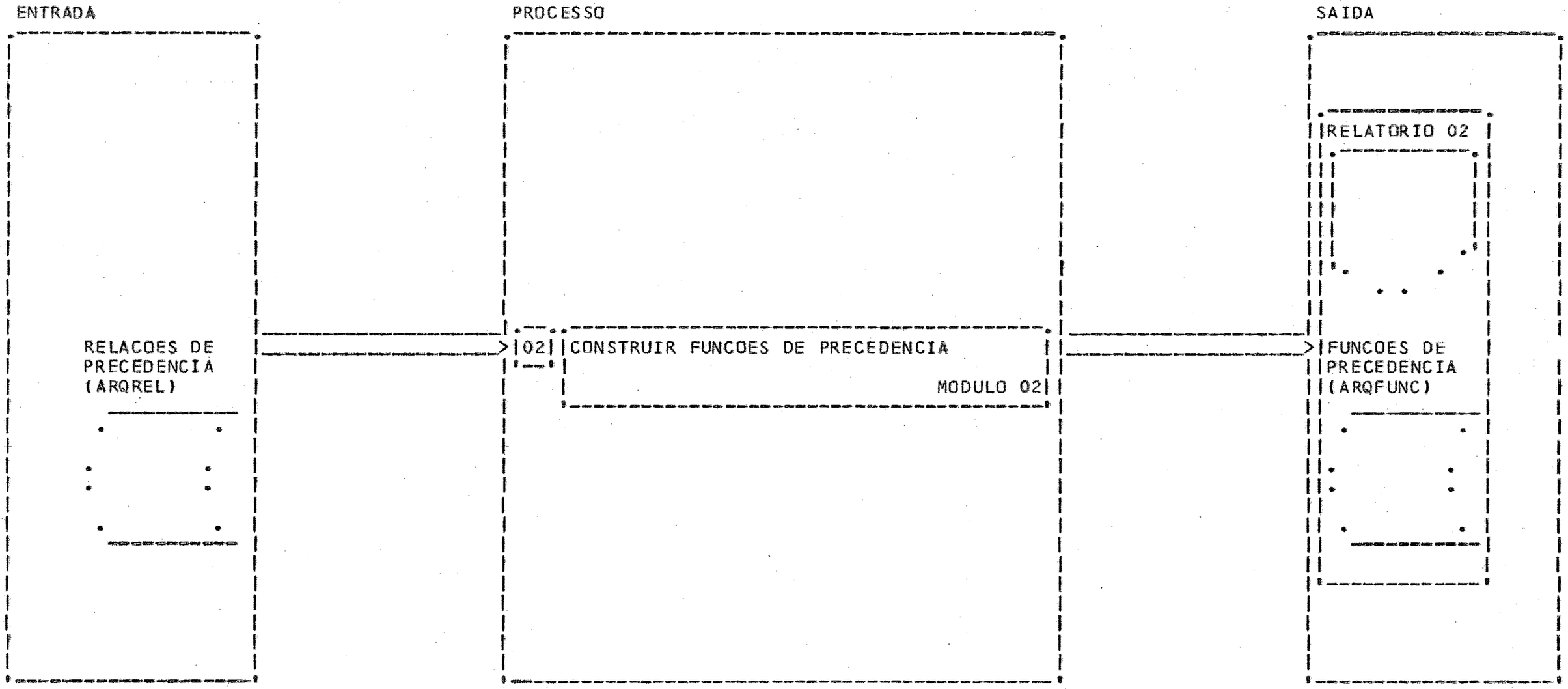


FIGURA III.4

A N A G R A M A - ANALISADOR DE GRAMATICAS DE PRECEDENCIA (CONT.)

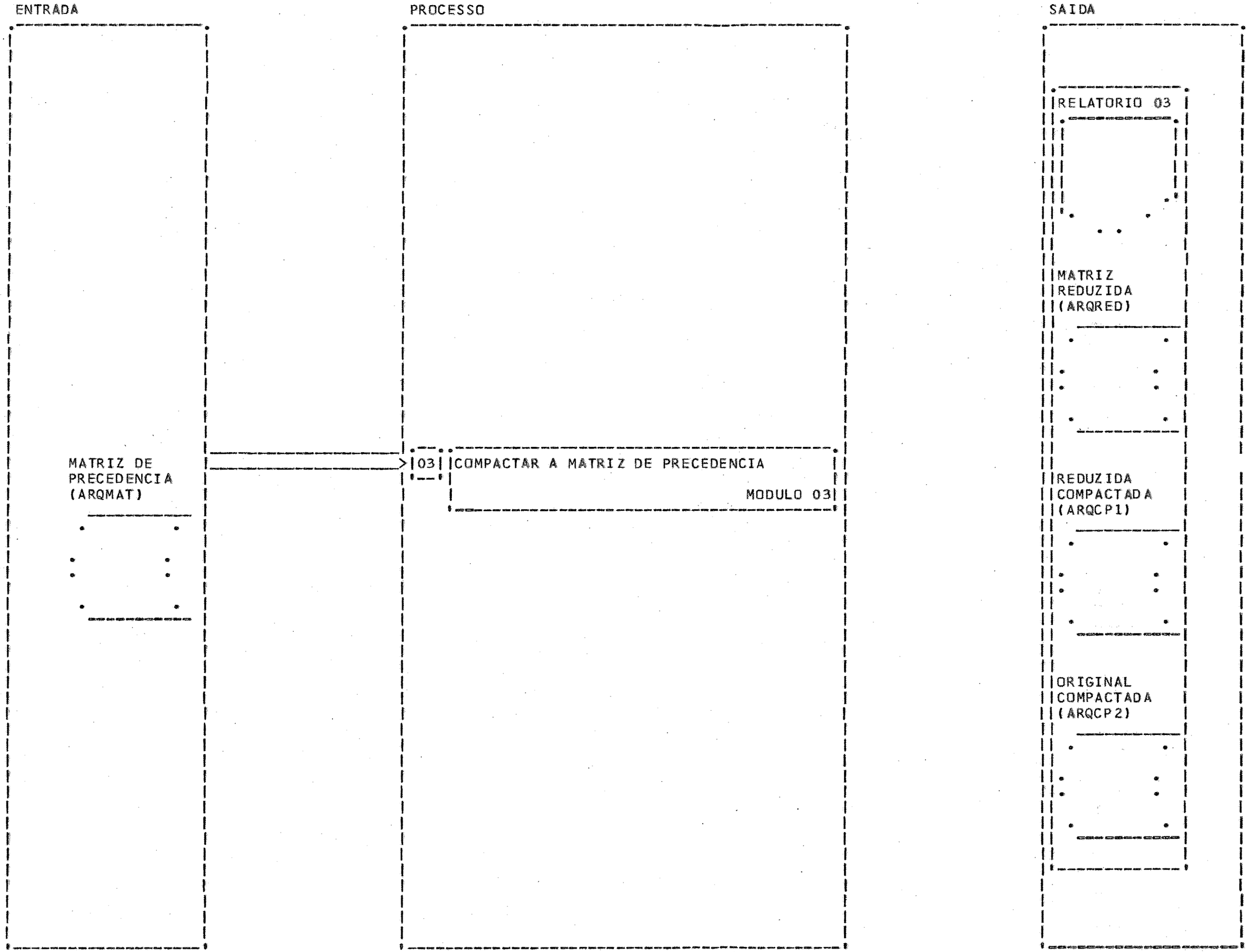


FIGURA III.5

c) cada símbolo da produção é constituído de uma sequência de no máximo 32 caracteres não brancos;

d) os símbolos são separados entre si por um ou mais brancos;

e) o símbolo raiz da produção deve necessariamente começar na posição 1;

f) o branqueamento das primeiras posições do registro indica que a raiz da produção é a raiz da produção anterior.

(2) comentários e registros em brancos:

registros com asterisco na posição 1 são considerados comentários; são permitidos registros totalmente em brancos para efeito de documentação e edição da BNF nos relatórios de saída.

(3) características físicas do arquivo:

os registros são imagens de cartão, tamanho fixo, 80 caracteres, gravados sequencialmente, constituindo um arquivo em disco de nome lógico (DDNAME para o OS) BNFONTE; este arquivo pode ser gerado a partir de cartões através do utilitário IEBGENER.

III.6 - Formato das Saídas

Entre todas as saídas emitidas por cada módulo, algumas são arquivos temporários, não sendo consideradas como saídas finais do ANAGRAMA.

Estes arquivos temporários são resultados intermediários gerados pelo Módulo 01 e utilizados como entrada pelos Módulos 02 e 03 (ver Figs. III.3, 4 e 5). As saídas efetivamente consideradas para o ANAGRAMA são os relatórios e as tabelas utilizadas pelo reconhecedor.

III.6.1 - Modelo dos Relatórios

Como foi dito, o sistema emite três relatórios. As Figs. III.6, 7, 8, 9 e 10 mostram folhas típicas de cada parte de que se compõe o primeiro relatório. As Figs. III.11 e 12 ilustram respectivamente os Relatórios 02 e 03.

III.6.2 - Arquivo das Produções Codificadas

a) nome lógico (DDNAME do OS): ARQPROD;

b) tamanho do registro: 80 caracteres;

c) conteúdo dos registros: cada símbolo da gramática é codificada como um inteiro; deste modo cada produção é gravada como uma sequência de inteiros; após cada sequência existe uma marca de fim de produção (0 zero) e em seguida o tamanho acumulado dos argumentos; este conjunto produção, marca e tamanho acumulado do argumento se repete para todas as produções; o fim do arquivo é assinalado por um zero na posição após o tamanho acumulado, ou seja, na posição da raiz de uma produção.

III.6.3 - Arquivo das Funções de Precedência

a) nome lógico (DDNAME do OS): ARQFUNC;

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

```
**********************************************************************************************************
* SEQUENCIAL *                    PRODUCOES DA GRAMATICA                                                 *
**********************************************************************************************************
```

EXEMPLO DE GRAMATICA DE PRECEDENCIA SIMPLES

(TIRADA DO GRIES, PG. 104, CAP.5)

```
001     <SIMBOLO_INICIAL>  ::=  B  <GRUPO_DE_A>  B
002     <GRUPO_DE_A>  ::=  (  <SUB_GRUPO_DE_A>
003                     |  A
004     <SUB_GRUPO_DE_A>  ::=  <GRUPO_DE_A>  A  )
```

FIGURA III.6

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SEQUENCIAL	RELACAO DOS NAO-TERMINAIS	PRODUCOES ONDE O SIMBOLO OCORRE
001	<SIMBOLO_INICIAL>	1
002	<GRUPO_DE_A>	1, 2, 3, 4
003	<SUB_GRUPO_DE_A>	2, 4

FIGURA III.7

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SEQUENCIAL	RELACAO DOS TERMINAIS	PRODUCOES ONDE O SIMBOLO OCORRE
004	B	1, 1
005	(	2
006	A	3, 4
007	)	4
008	ZZ	*** SIMBOLO CRIADO COMO DELIMITADOR DE CADEIA ***

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SIMBOLO		VIZINHANCA A DIREITA		PRECEDENCIA NO PAR
1	<SIMBOLO_INICIAL>	8	ZZ	>
2	<GRUPO_DE_A>	4	B	=
2	<GRUPO_DE_A>	6	A	=
3	<SUB_GRUPO_DE_A>	4	B	>
3	<SUB_GRUPO_DE_A>	6	A	>
4	B	2	<GRUPO_DE_A>	=
4	B	5	(	<
4	B	6	A	<
4	B	8	ZZ	>
5	(	2	<GRUPO_DE_A>	<
5	(	3	<SUB_GRUPO_DE_A>	=
5	(	5	(	<
5	(	6	A	<
6	A	4	B	>
6	A	6	A	>
6	A	7	)	=
7	)	4	B	>
7	)	6	A	>
8	ZZ	1	<SIMBOLO_INICIAL>	<
8	ZZ	4	B	<

FIGURA III.9

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

****
* DIAGNOSTICOS E MENSAGENS DE ERRO *
****

A GRAMATICA E' DE PRECEDENCIA SIMPLES
- SIMBOLO INICIAL DA GRAMATICA: <SIMBOLO_INICIAL>
- TAMANHO ACUMULADO DOS ARGUMENTOS: 9

FIGURA III.10

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 02 - 13-SET-79

****
* CARACTERISTICAS DA GRAMATICA E FUNCOES LINEARES DE PRECEDENCIA *
****

CARACTERISTICAS DA GRAMATICA

- NUMERO DE PRODUCOES: 4
- NUMERO DE TERMINAIS: 5
- NUMERO DE NAO TERMINAIS: 3
- TAMANHO ACUMULADO DOS ARGUMENTOS: 9
- TIPO DE PRECEDENCIA: SIMPLES

VALORES DAS FUNCOES LINEARES

FUNCAO F :
F(1)= 1, F(2)= 2, F(3)= 3, F(4)= 1, F(5)= 0, F(6)= 3, F(7)= 3
F(8)= 0

FUNCAO G :
G(1)= 1, G(2)= 1, G(3)= 0, G(4)= 2, G(5)= 2, G(6)= 2, G(7)= 3
G(8)= 0

FIGURA III.11

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 03 - 13-SET-79

****
* RESULTADOS DA COMPACTACAO DA MATRIZ DE PRECEDENCIA *
****

CARACTERISTICAS DA GRAMATICA

- NUMERO DE PRODUCOES: 4
- NUMERO DE TERMINAIS: 5
- NUMERO DE NAO TERMINAIS: 3
- TAMANHO ACUMULADO DOS ARGUMENTOS: 9
- TIPO DE PRECEDENCIA: SIMPLES

RESULTADOS DA COMPACTACAO

MATRIZ ORIGINAL : 8 LINHAS E 8 COLUNAS, OCUPANDO 17 CARACTERES.

MATRIZ REDUZIDA : 7 LINHAS, 8 COLUNAS, DOIS VETORES INDEXADORES DE LINHAS E COLUNAS, OCUPANDO TUDO 30 CARACTERES, COM -76.47% DE COMPACTACAO.

MATRIZ ORIGINAL COMPACTADA : 39 PARES (Y, PRECEDENCIA), 8 PONTEIROS DE LINHA, OCUPANDO TUDO 66 CARACTERES, COM -286.76% DE COMPACTACAO.

MATRIZ REDUZIDA COMPACTADA : 33 PARES (Y, PRECEDENCIA), 7 PONTEIROS DE LINHA, 8 INDEXADORES DE COLUNAS, OCUPANDO TUDO 66 CARACTERES, COM -289.71% DE COMPACTACAO.

FIGURA III.12

b) tamanho do registro: 80 caracteres;

c) conteúdo do primeiro registro: aqui são gravados os controles da gramática a serem usados pelo reconhecedor; cada controle é gravado no formato BIN FIXED (15,0) do PL/I, a partir do terceiro byte; os controles são os seguintes, nesta ordem: número de não-terminais, número de símbolos, número de produções, tamanho total dos argumentos, código do símbolo inicial e indicador de precedência (0 indica precedência simples e 1 índice precedência fraca);

d) conteúdo dos demais registros: são gravadas sequencialmente os valores das duas funções, cada valor ocupando um byte, tomando tudo tantos registros quantos necessários.

III.6.4 - Arquivo da Matriz Reduzida

a) nome lógico (DDNAME do OS): ARQRED

b) tamanho do registro: 80 caracteres;

c) conteúdo do primeiro registro: registro de controle como descrito para o arquivo das funções de precedência;

d) matriz reduzida: gravada linha a linha e cada valor ocupa 2 bits, tomando tudo tantos registros quantos necessários;

f) índices redutores: gravados sequencialmente a partir do primeiro registro em seguida aquele onde a matriz reduzida terminou de ser gravada; primeiro é gravado o índice redutor de linha, em seguida o redutor de colunas; cada valor ocupa 1 byte, ocupando tudo tantos registros quantos ne-

cessários.

III.6.5 - Arquivo da Matriz Original Compactada

a) nome lógico (DDNAME do OS): ARQCP1;

b) tamanho do registro: 80 caracteres;

c) conteúdo do primeiro registro: registro de controle como descrito para o arquivo das funções de precedência:

d) sequência de pares: cada registro contém dois vetores, o primeiro contendo os primeiros elementos dos pares e o segundo contendo os segundos elementos; cada valor do primeiro vetor ocupa 1 byte e cada valor do segundo vetor ocupa 2 bits; portanto, cada registro deste arquivo tem 30 ocorrências de cada um destes vetores; a sequência de pares ocupará tantos registros quantos necessários;

e) vetor indexador dos pares: gravado a partir do primeiro registro em seguida aquele onde a sequência de pares terminou de ser gravada; cada valor do indexador é gravado no formato BIN FIXED (15,0) do PL/I, tomando tudo tantos registros quantos necessários.

III.6.6 - Arquivo da Matriz Reduzida Compactada

a) nome lógico (DDNAME do OS): ARQCP2;

b) tamanho do registro: 80 caracteres;

c) conteúdo do primeiro registro: registro de controle como descrito para as funções de precedência;

d) sequência de pares: igual como descrito para a matriz original compactada;

e) a partir do primeiro registro em seguida aquele onde a sequência de pares terminou de ser gravada, começa a gravação do vetor indexador de linha; cada valor deste vetor é gravado no formato BIN FIXED (15,0) do PL/I; em seguida é gravado o vetor redutor de colunas e cada valor ocupa 1 byte; estes dois vetores juntos ocupam tantos registros quantos necessários.

III.6.7 - Arquivo da Matriz Original

a) nome lógico (DDNAME do OS): ARQPREC;

b) tamanho do registro: 80 caracteres;

c) conteúdo do primeiro registro: controles, como descrito para o arquivo das funções de precedência;

d) matriz de precedência: gravada linha a linha e cada valor ocupa 2 bits, tomando tudo tantos registros quantos necessários.

III.7 - Implementação do Sistema

O sistema foi implementado no OS/IBM e cada módulo constitue um JOB distinto. A seguir é mostrado um exemplo de registros JCL para os três módulos do ANAGRAMA.

```
         ********************************************
         *  EXEMPLO DE REGISTROS JCL PARA O MODULO 01  *
         ********************************************


//P422470A JOB P4010438,BENEH,CLASS=J,MSGCLASS=D
//*************************************************************************
//********************  ANAGRAMA  -  MODULO 01  ***************************
//********************* ANALISAR PRODUCOES DE ENTRADA *********************
//*************************************************************************
//GRAVAR EXEC PGM=IEBGENER
//SYSUT2 DD DSN=&&BNFONTE,UNIT=SYSDA,SPACE=(TRK,(10,1)),
// DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSPRINT DD SYSOUT=D
//SYSIN DD DUMMY
//SYSUT1 DD *
***************    AQUI ENTRAM AS PRODUCOES CODIFICADAS  ****************
/*
//ANAG101 EXEC PLIXCLG,
//   PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//*************************************************************************
//*                    CODIFICAR SIMBOLOS E PRODUCOES                     *
//*************************************************************************
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
********************* AQUI ENTRA O PROGRAMA ANAG101 ***********************
/*
//LKED.SYSLMOD DD DSN=&&GOSET(GOX)
//LKED.SYSPRINT DD SYSOUT=D
//GO.BNFONTE DD DSN=&&BNFONTE,DISP=(OLD,PASS)
//GO.ARQPROD DD DSN=ANAGRAMA.ARQPROD,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//    VOL=SER=VS1003,DISP=(NEW,PASS,DELETE),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.ARQTAB DD DSN=&&ARQTAB,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//    VOL=SER=VS1003,DISP=(NEW,PASS,DELETE),
//    DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ANAG102 EXEC PLIXCLG,
//   PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//*************************************************************************
//*                      GERAR MATRIZES PARCIAIS                          *
//*************************************************************************
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
***************  AQUI ENTRA O PROGRAMA ANAG102  ***************************
/*
//LKED.SYSLMOD DD DSN=&&GOSET(GOY)
//LKED.SYSPRINT DD SYSOUT=D
//GO.ARQTAB DD DSN=&&ARQTAB,DISP=(OLD,PASS)
//GO.ARQPROD DD DSN=ANAGRAMA.ARQPROD,DISP=(OLD,PASS)
//GO.ARQOCOR DD DSN=&&ARQOCOR,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//  VOL=SER=VS1003,DISP=(NEW,PASS,DELETE),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.ARQCTL1 DD DSN=&&ARQCTL1,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//  DISP=(NEW,PASS,DELETE),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.ARQREL DD DSN=ANAGRAMA.ARQREL,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//    VOL=SER=VS1003,DISP=(NEW,PASS,DELETE),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
```

```
//ANAG103 EXEC PLIXCLG,
//   PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//*****************************************************************
//*                  DIAGNOSTICAR TIPO DE PRECEDENCIA             *
//*****************************************************************
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
**************  AQUI ENTRA O PROGRAMA ANAG103  ***********************
/*
//LKED.SYSLMOD DD DSN=&&GOSET(GOW)
//LKED.SYSPRINT DD SYSOUT=D
//GO.ARQCTL1 DD DSN=&&ARQCTL1,DISP=(OLD,PASS)
//GO.ARQREL DD DSN=ANAGRAMA.ARQREL,DISP=(OLD,KEEP,DELETE)
//GO.ARQOCOR DD DSN=&&ARQOCOR,DISP=(OLD,PASS)
//GO.ARQPROD DD DSN=ANAGRAMA.ARQPROD,DISP=(OLD,DELETE)
//GO.ARQMAT DD DSN=ANAGRAMA.ARQMAT,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//    VOL=SER=VS1003,DISP=(NEW,PASS),
//    DCB=(RECFM=F,LRECL=256,BLKSIZE=256,DSORG=DA,OPTCD=R)
//GO.ARQCTL2 DD DSN=ANAGRAMA.ARQCTL2,UNIT=SYSDA,SPACE=(TRK,(10,1)),
//  DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//  VOL=SER=VS1003
//ANAG104 EXEC PLIXCLG,
//   PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//*****************************************************************
//*                    EMITIR PRIMEIRO RELATORIO                  *
//*****************************************************************
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
**************   AQUI ENTRA O PROGRAMA ANAG104  ********************
/*
//LKED.SYSLMOD DD DSN=&&GOSET(GOZ)
//LKED.SYSPRINT DD SYSOUT=D
//GO.SYSPRINT DD SYSOUT=D
//GO.BNFONTE DD DSN=&&BNFONTE,DISP=(OLD,PASS)
//GO.ARQCTL2 DD DSN=ANAGRAMA.ARQCTL2,DISP=(OLD,KEEP,DELETE)
//GO.ARQTAB DD DSN=&&ARQTAB,DISP=(OLD,PASS)
//GO.ARQOCOR DD DSN=&&ARQOCOR,DISP=(OLD,PASS)
//GO.ARQMAT DD DSN=ANAGRAMA.ARQMAT,DISP=(OLD,KEEP,DELETE)
//
```

```
         ************************************************************
         *  EXEMPLO DE REGISTROS JCL PARA OS MODULOS 02 E 03  *
         ************************************************************

//P422470B JOB P4010438,BENEH,CLASS=J,MSGCLASS=D
//*************************************************************
//**************** ANAGRAMA  (MODULO 02) **********************
//********** GERAR FUNCOES LINEARES DE PRECEDENCIA ************
//*************************************************************
//ANAG201 EXEC PLIXCLG,
//  PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
***************  AQUI ENTRA O PROGRAMA ANAG201  ***************************
/*
//LKED.SYSPRINT DD SYSOUT=D
//GO.SYSPRINT DD SYSOUT=D
//GO.ARQCTL2 DD DSN=ANAGRAMA.ARQCTL2,DISP=(OLD,KEEP),
//   VOL=SER=VS1003,UNIT=SYSDA
//GO.ARQREL DD DSN=ANAGRAMA.ARQREL,DISP=(OLD,KEEP),
//   VOL=SER=VS1003,UNIT=SYSDA
//GO.ARQFUNC DD DSN=ANAGRAMA.ARQFUNC,DISP=(NEW,DELETE),SPACE=(TRK,1),
//   UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//



//P422470C JOB P4010438,BENEH,CLASS=J,MSGCLASS=D
//*************************************************************
//*********************  ANAGRAMA   (MODULO 03) ***************
//****************** COMPACTAR MATRIZ DE PRECEDENCIA **********
//*************************************************************
//ANAG301 EXEC PLIXCLG,
//  PARM.PLI='NS,CS(48),F(I)',PARM.LKED=
//PLI.SYSPRINT DD SYSOUT=D
//PLI.SYSIN DD *
******************  AQUI ENTRA O PROGRAMA ANAG301 ********************
/*
//LKED.SYSPRINT DD SYSOUT=D
//GO.SYSPRINT DD SYSOUT=D
//GO.ARQCTL2 DD DSN=ANAGRAMA.ARQCTL2,DISP=(OLD,KEEP),
//   VOL=SER=VS1003,UNIT=SYSDA
//GO.ARQMAT DD DSN=ANAGRAMA.ARQMAT,DISP=(OLD,KEEP),
//   DCB=(DSORG=DA,OPTCD=R),
//   VOL=SER=VS1003,UNIT=SYSDA
//GO.ARQRED DD DSN=&&ARQRED,DISP=(NEW,PASS),SPACE=(TRK,1),
//  UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=80,RECFM=FB)
//GO.ARQCP1 DD DSN=&&ARQCP1,DISP=(NEW,PASS),SPACE=(TRK,1),
//  UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=80,RECFM=FB)
//GO.ARQCP2 DD DSN=&&ARQCP2,DISP=(NEW,PASS),SPACE=(TRK,1),
//   UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=80,RECFM=FB)
//GO.ARQPREC DD DSN=&&ARQPREC,DISP=(NEW,PASS),SPACE=(TRK,1),
//   UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=80,RECFM=FB)
//
```

Com A de ave se escreve amor e arma.

Com B de bola se escreve belo e bala.

Com C de casa se escreve céu e cela.

"ABC"

FAUSTO NILO, com música de Fagner

CAPÍTULO IV

ANALISAR PRODUÇÕES DE ENTRADA

O Módulo 01 do ANAGRAMA analisa as produções de entrada e identifica os elementos e características da gramática com base nos conceitos de gramáticas livre do contexto e, em particular, nos conceitos de precedência simples e fraca. Desta maneira, são extraidos os terminais, os não-terminais, o símbolo inicial, as relações de precedência existente entre os símbolos e também fornecido o diagnóstico final sobre o tipo de precedência da gramática sendo analisada. Todos estes resultados constam de um relatório final emitido pelo Módulo e, adicionalmente, são gravados em disco alguns arquivos para uso dos outros módulos. Um destes arquivos é a codificação das produções, já pronta para uso também em um reconhecedor sintático.

IV.1 - Entrada do Módulo

A gramática a ser analisada é representada pelo seu conjunto de produções P que é assim o dado inicial para a análise a ser feita pelo módulo.

IV.2 - Saídas do Módulo

A partir das produções são obtidas cinco saídas, a saber:

- os outros três elementos N, T, S, definidores da gramática G;

ANAGRAMA - MODULO 01 - ANALISAR PRODUCOES DE ENTRADA

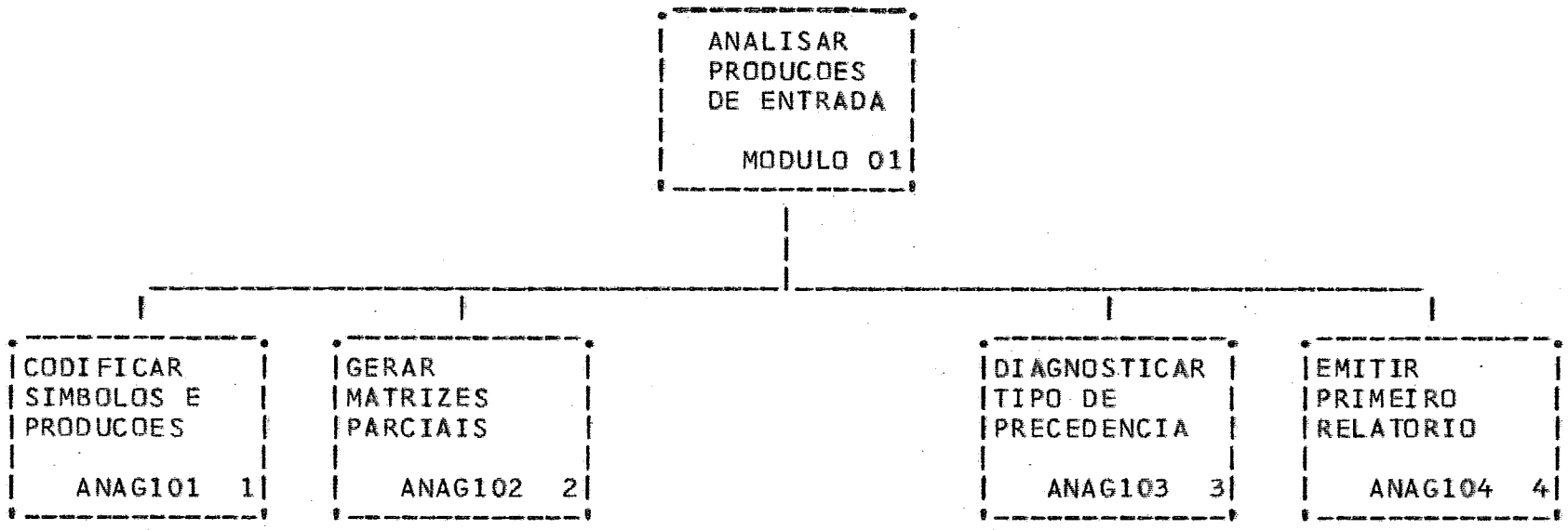
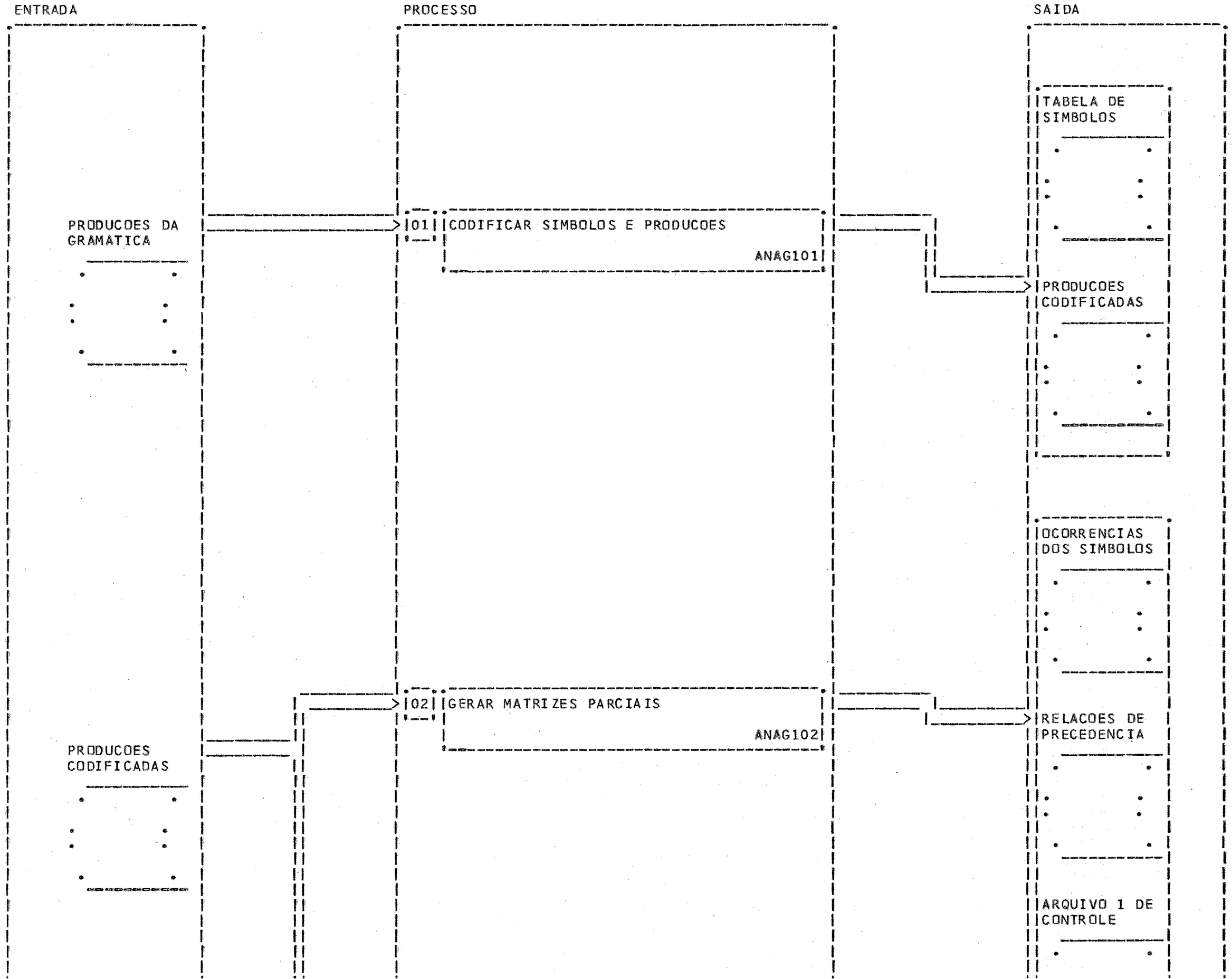


FIGURA IV.1

ANAGRAMA - MODULO 01 - ANALISAR PRODUCOES DE ENTRADA



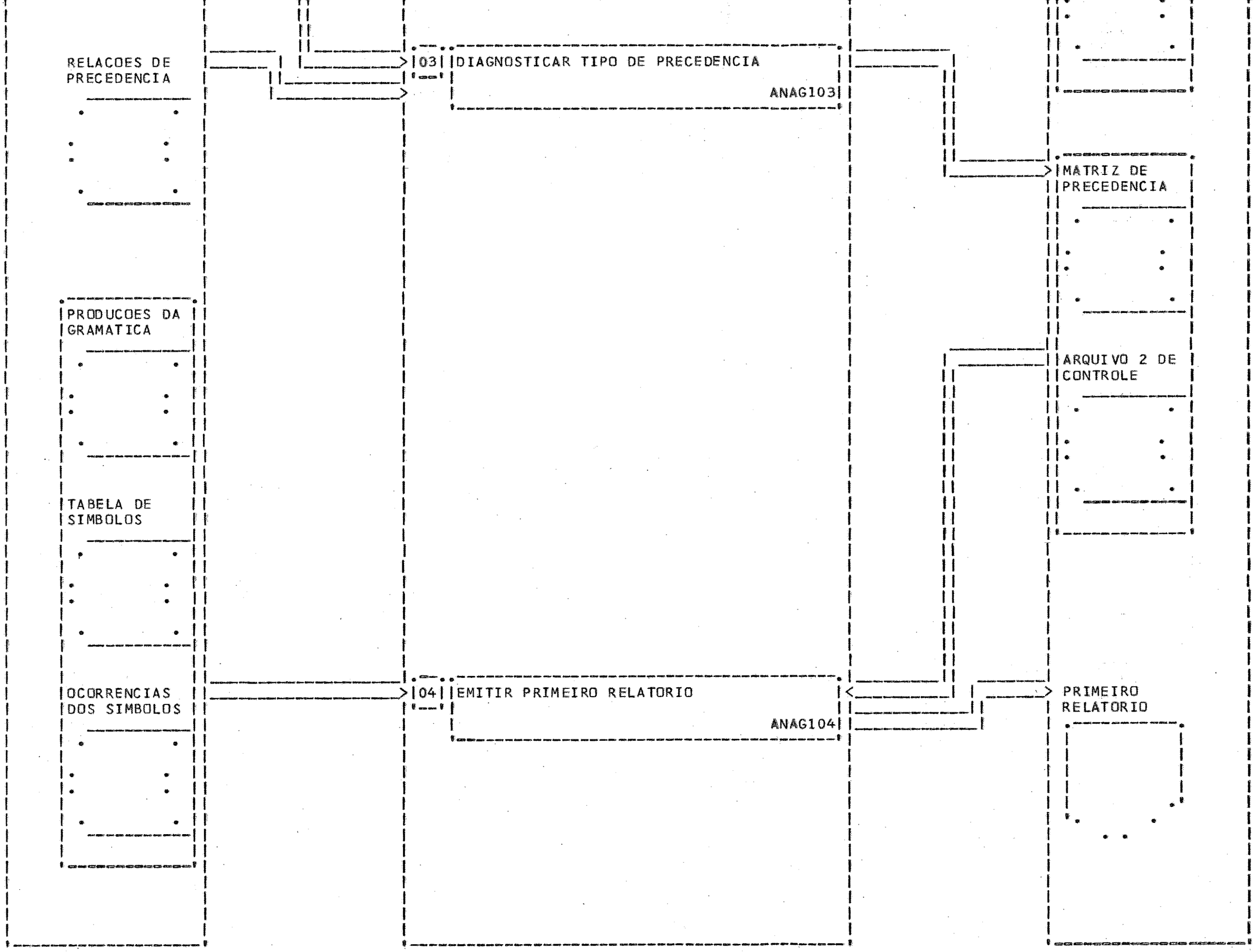

RELACOES DE PRECEDENCIA
03
DIAGNOSTICAR TIPO DE PRECEDENCIA
ANAG103
MATRIZ DE PRECEDENCIA
PRODUCOES DA GRAMATICA
ARQUIVO 2 DE CONTROLE
TABELA DE SIMBOLOS
OCORRENCIAS DOS SIMBOLOS
04
EMITIR PRIMEIRO RELATORIO
ANAG104
PRIMEIRO RELATORIO

FIGURA IV.2

- as relações de precedência existentes sobre os símbolos de G, construídas na forma de um conjunto de triplas (s_i, s_j, r), onde s_i e s_j são símbolos de G e r o tipo de relação entre estes símbolos;

- a matriz de precedência M com n linhas e m colunas;

- o tipo de precedência da gramática G;

- forma codificada das produções P, a ser usada pelo analisador sintático.

IV.3 - Funções do Módulo

Como mostram as figuras IV.1 e 2, o Módulo Ø1 se compõe básicamente de quatro funções, cada uma delas descrita a seguir.

IV.3.1 - Codificar Símbolos e Produções

Aqui são determinados os conjuntos N e T, não-terminais e terminais, da gramática. A definição do que seja não-terminal ou terminal é feita segundo o símbolo seja ou não raiz de uma produção. Estes dois conjuntos definem uma tabela de símbolos que é gravada em disco. Cada símbolo da gramática é codificado segundo um inteiro e as produções já utilizando estes códigos são gravadas em disco. Esta função grava assim dois arquivos em disco, sendo que a tabela de símbolos é um arquivo temporário usado apenas na função que imprime o relatório. São determinados o número de terminais, de não-terminais e o tamanho acumulado dos argumentos das produções, gravados junto com a tabela de símbolo em um registro de controle da gramática. Erros no formato das produções de entrada são tam-

bém assinalados nestes controles.

IV.3.2 - Gerar Matrizes Parciais

A partir das produções já codificadas são construídas as relações PRIMEIRO, ÚLTIMO e IGUAL, com base nas definições apresentadas no Capítulo II. Em seguidas as relações MENOR e MAIOR são construídas e juntamente com a relação IGUAL, são gravadas como um conjunto de triplas (s_i, s_j, r), onde s_i, s_j são símbolos de G e r indica a relação entre eles. Este conjunto de triplas representando as relações de precedência constitue um arquivo em disco utilizado pelo Módulo 02. O símbolo delimitador de cadeias é criado e acrescentado ao conjunto terminal da gramática (ver Capítulo II, item 16, a conceituação de símbolo delimitador de cadeia).

Esta função determina para cada símbolo de G em quais produções este símbolo ocorre. Esta sequência de ocorrências para cada símbolo constitue um arquivo temporário, gravado em disco, e usado pela função que imprime o primeiro relatório. Finalmente, há um terceiro arquivo contendo apenas os controles, aos quais é acrescentado o símbolo inicial S da gramática, que é determinado nesta função. Estes controles poderão conter também indicação de erros acontecidos nesta ou na função anterior.

A seguir, os dois algoritmos utilizados na construção das relações MENOR e MAIOR.

Algoritmo M - Construir as Relações MENOR e MAIOR

Passo 1: Fazer as relações MENOR e MAIOR inicialmente vazias.
Os pares seguintes são sucessivamente executados para todos os símbolos s de G.

Passo 2: Fazer os conjuntos ESQUERDOS e DIREITOS vazios;
obter os não-terminais A, tal que (s, A) $\in$ IGUAL;
utilizando o Algoritmo T, para cada A construir os conjuntos
$E = \{ v \mid (A,v) \in PRIMEIRO^{+} \}$,
$D = \{ v \mid (A,v) \in ÚLTIMO^{+} \}$,
e em seguida fazer
ESQUERDOS = ESQUERDOS $\cup$ E,
DIREITOS = DIREITOS $\cup$ D.

Passo 3: Gerar a relação MENOR fazendo
M = {s} . ESQUERDOS e
MENOR = MENOR $\cup$ M.

Passo 4: Obter todos os símbolos v tal que (s, v) $\in$ IGUAL
e para cada v fazer
ESQUERDOS = ESQUERDOS $\cup$ {v};
gerar a relação MAIOR fazendo
M = DIREITOS . ESQUERDOS e
MAIOR = MAIOR $\cup$ M.

Algoritmo T - Construir Fechamento Transitivo

Dada uma relação R, este algoritmo efetua o fechamento transitivo R^+ para um símbolo dado A, gerando o conjunto

EXTREMOS = EXTREMOS $\cup$ { v | (A,v) $\in R^+$ }

A relação R pode ser PRIMEIRO ou ÚLTIMO e, consequentemente, EXTREMOS pode ser ESQUERDOS ou DIREITOS.

Passo 1: Para todo v, tal que (A, v) $\in$ R fazer
EXTREMOS = EXTREMOS $\cup$ {v} .

Passo 2: Seja B $\in$ EXTREMOS;
para todo v tal que (B,v) $\in$ R fazer
EXTREMOS = EXTREMOS $\cup$ {v}
repetir este passo até o último elemento de EXTREMOS.

IV.3.3 - Diagnosticar Tipo de Precedência

A partir das três relações de precedência geradas na função anterior esta terceira função constroi a matriz de precedência detectando e registrando todos os conflitos de precedência existentes entre dois símbolos. É verificado se a gramática é de precedência simples. Se os conflitos existentes forem apenas do tipo MENOR - IGUAL são testadas as duas condições da definição de precedência fraca. O diagnóstico final resultante destas verificações é codificado segundo um indicador de precedência e gravado no arquivo de controles. Este indicador de precedência será interpretado sob a forma de mensagens e diagnósticos emitidas pelo Relatório 01. A

matriz de precedência M com n linhas e m colunas é também gravada em disco para ser usada pela função que imprime o primeiro relatório e pelo Módulo 03 de compactação.

IV.3.4 - Emitir Primeiro Relatório

A partir dos arquivos gerados nas três funções anteriores é impresso o Relatório 01 cujas folhas modelos já foram mostradas no Capítulo III.

IV.4 - Implementação do Módulo

Cada função do Módulo 01 é implementado como um programa. Deste modo existem quatro programas respectivamente de nomes ANAG101, ANAG102, ANAG103 e ANAG104. As figuras seguintes deste capítulo documentam o módulo como um todo e cada programa em particular. Finalmente, é fornecida a listagem em imagem de cartões destes quatro programas.

Finalmente, vale lembrar que os recursos de memória são alocados em função da gramática sendo analisada, de modo que problemas de estouro de memória devem ser analisados de acordo com cada instalação onde o Módulo estiver implantado.

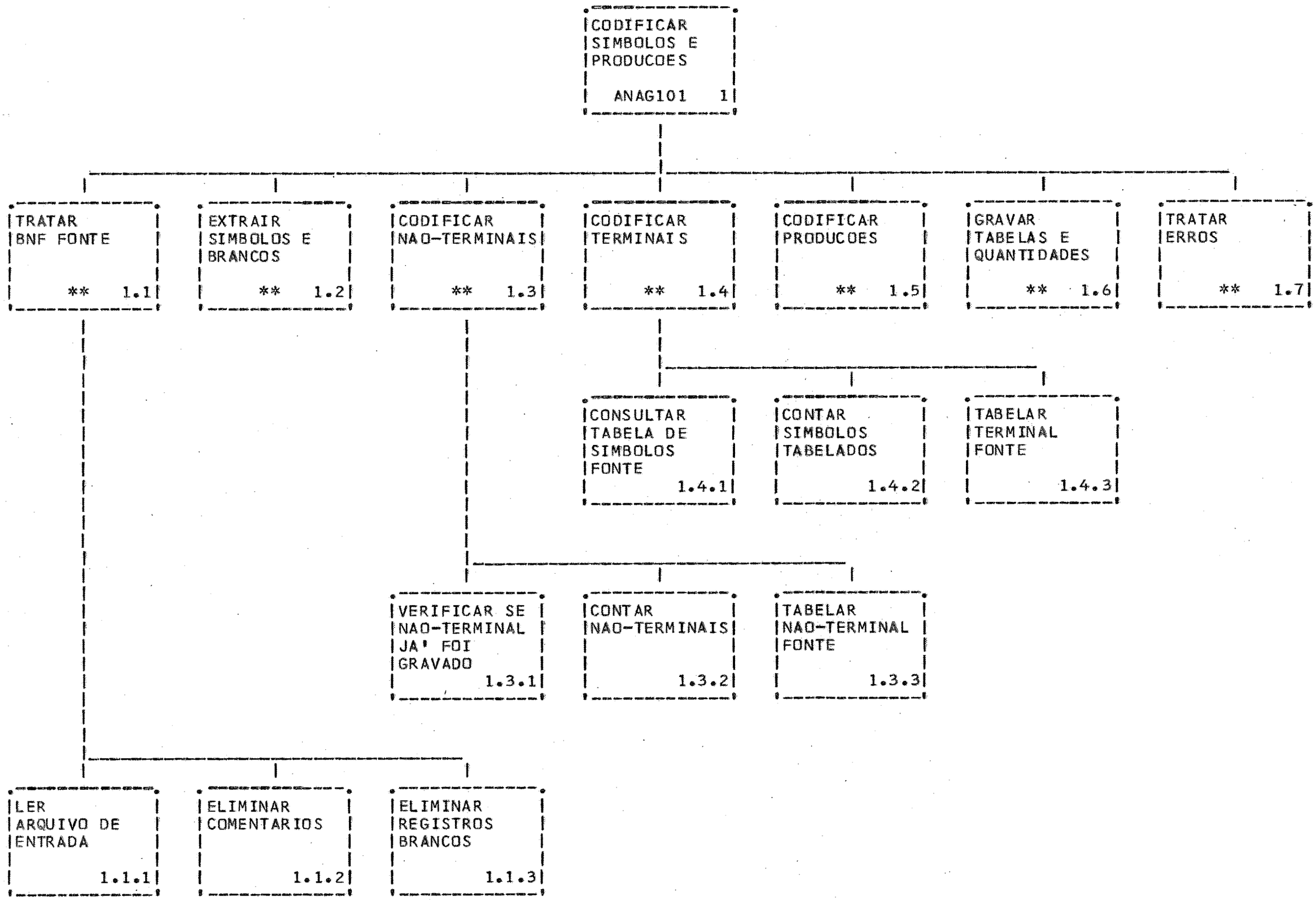

CODIFICAR SIMBOLOS E PRODUCOES
ANAG101 1
TRATAR BNF FONTE
** 1.1
EXTRAIR SIMBOLOS E BRANCOS
** 1.2
CODIFICAR NAO-TERMINAIS
** 1.3
CODIFICAR TERMINAIS
** 1.4
CODIFICAR PRODUCOES
** 1.5
GRAVAR TABELAS E QUANTIDADES
** 1.6
TRATAR ERROS
** 1.7
CONSULTAR TABELA DE SIMBOLOS FONTE
1.4.1
CONTAR SIMBOLOS TABELADOS
1.4.2
TABELAR TERMINAL FONTE
1.4.3
VERIFICAR SE NAO-TERMINAL JA' FOI GRAVADO
1.3.1
CONTAR NAO-TERMINAIS
1.3.2
TABELAR NAO-TERMINAL FONTE
1.3.3
LER ARQUIVO DE ENTRADA
1.1.1
ELIMINAR COMENTARIOS
1.1.2
ELIMINAR REGISTROS BRANCOS
1.1.3

FIGURA IV.3

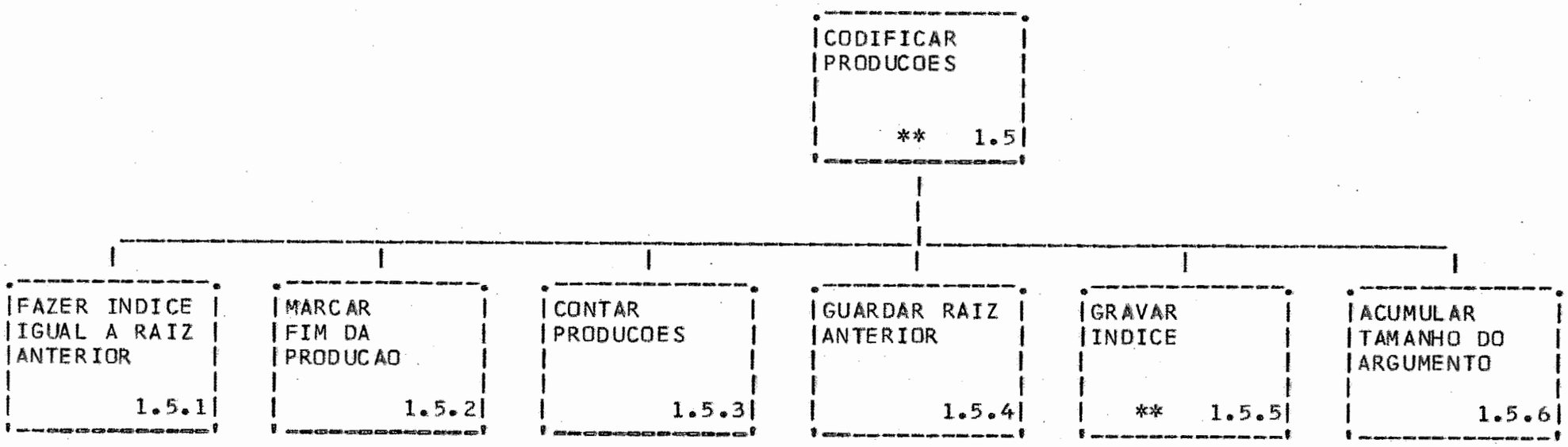

CODIFICAR PRODUCOES
** 1.5
FAZER INDICE IGUAL A RAIZ ANTERIOR
1.5.1
MARCAR FIM DA PRODUCAO
1.5.2
CONTAR PRODUCOES
1.5.3
GUARDAR RAIZ ANTERIOR
1.5.4
GRAVAR INDICE
** 1.5.5
ACUMULAR TAMANHO DO ARGUMENTO
1.5.6

FIGURA IV.4

ANAG101 - CODIFICAR SIMBOLOS E PRODUCOES (1)

ENTRADA

PRODUCOES DA GRAMATICA (BNFONTE)

PRODUCAO

SIMBOLO

TAMANHO

SIMBOLO

INDICE NA TABELA

TABELA DE SIMBOLOS

PROCESSO

01 TRATAR BNF FONTE.

EXTRAIR DO ARQUIVO DE ENTRADA UM REGISTRO CONTENDO UMA PRODUCAO. ELIMINA REGISTROS BRANCOS E COM COMENTARIOS.

02 EXTRAIR SIMBOLOS E BRANCOS.

PERCORRE UMA PRODUCAO A PARTIR DE UM CARATER COM POSICAO DADA POR 'PONT'. SE ESTE CARATER E' BRANCO O PROXIMO NAO BRANCO E' PROCURADO E O SIMBOLO EXTRAIDO TEM TAMANHO ZERO. CASO CONTRARIO O SIMBOLO E' EXTRAIDO E OS BRANCOS SEGUINTES SAO ELIMINADOS. 'PONT' SEMPRE APONTA UM CARATER NAO BRANCO AO FIM DESTA FUNCAO. SE TAMANHO>32 O SIMBOLO E' TRUNCADO.

03 CODIFICAR NAO TERMINAIS.

TODOS OS SIMBOLOS COMECANDO NO PRIMEIRO CARATER DE CADA PRODUCAO SAO TABELADOS COMO OS NAO TERMINAIS DA GRAMATICA. DESTA MANEIRA, EM UM PRIMEIRO CICLO, AS FUNCOES 1, 2 E 3 SAO SUCESSIVAMENTE ACIONADAS ATE' QUE OS NAO TERMINAIS FICAM OCUPANDO AS NAOTERMINAIS POSICOES INICIAS DA TABELA.

04 CODIFICAR TERMINAIS.

TABELADOS OS NAO TERMINAIS, TODOS OS SIMBOLOS DAS PRODUCOES SAO OBTIDOS E POR EXCLUSAO SAO TABELADOS OS TERMINAIS. SE O TAMANHO DO SIMBOLO E' NULO A FUNCAO NAO E' EXECUTADA . E' GERADO O INDICE NA TABELA DO SIMBOLO RECEBIDO. CADA NOVO SIMBOLO TABELADO INCREMENTA O CONTADOR 'NUMERO DE SIMBOLOS'.

SAIDA

PRODUCAO

PONT

TAMANHO

NAO TERMINAIS

TABELA DE SIMBOLOS

NUMERO DE SIMBOLOS

TABELA DE SIMBOLOS

INDICE NA TABELA

PRODUCOES CODIFICADAS (ARQPROD)

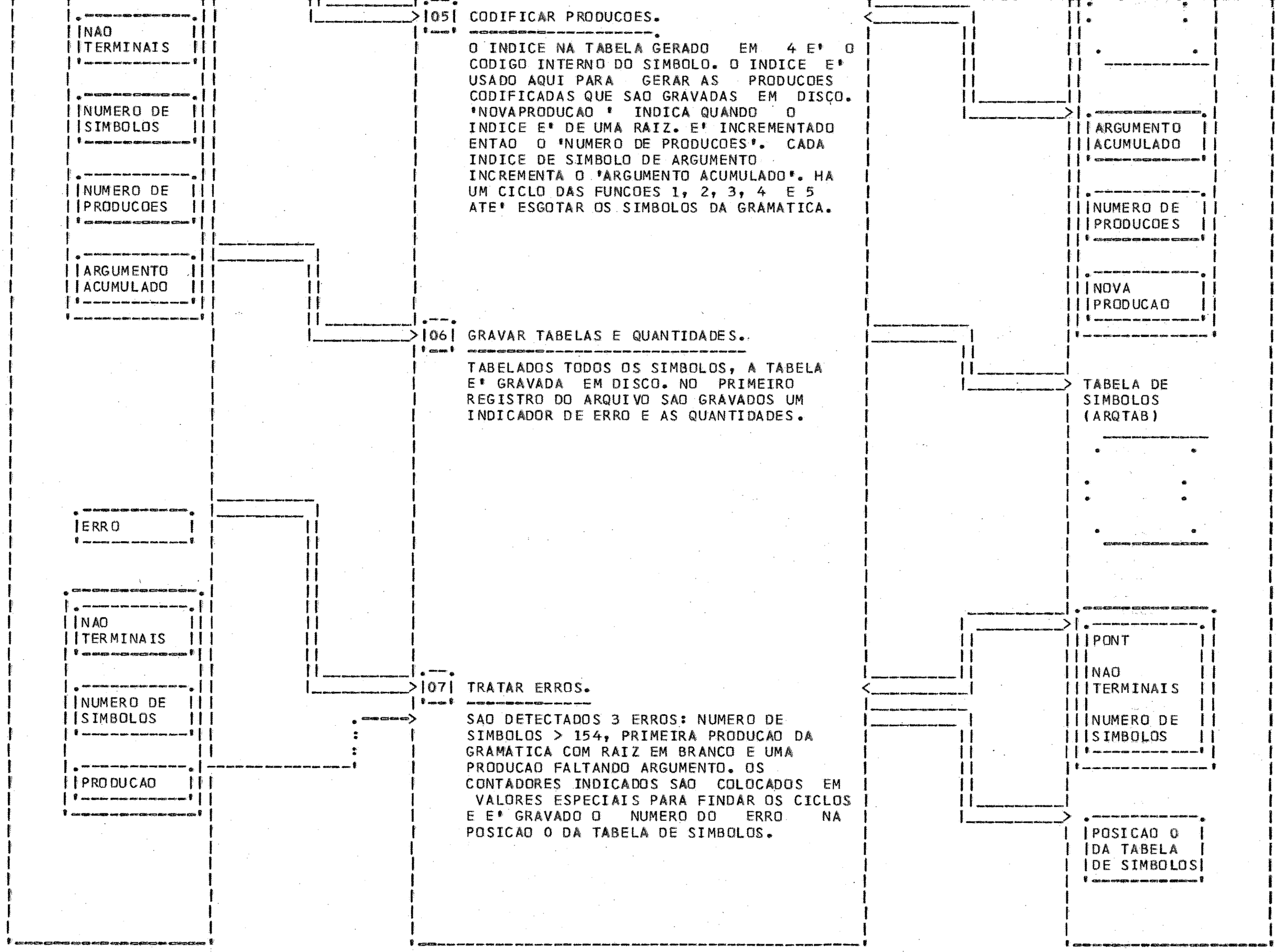

05 CODIFICAR PRODUCOES.
O INDICE NA TABELA GERADO EM 4 E' O CODIGO INTERNO DO SIMBOLO. O INDICE E' USADO AQUI PARA GERAR AS PRODUCOES CODIFICADAS QUE SAO GRAVADAS EM DISCO. 'NOVAPRODUCAO ' INDICA QUANDO O INDICE E' DE UMA RAIZ. E' INCREMENTADO ENTAO O 'NUMERO DE PRODUCOES'. CADA INDICE DE SIMBOLO DE ARGUMENTO INCREMENTA O 'ARGUMENTO ACUMULADO'. HA UM CICLO DAS FUNCOES 1, 2, 3, 4 E 5 ATE' ESGOTAR OS SIMBOLOS DA GRAMATICA.
06 GRAVAR TABELAS E QUANTIDADES.
TABELADOS TODOS OS SIMBOLOS, A TABELA E' GRAVADA EM DISCO. NO PRIMEIRO REGISTRO DO ARQUIVO SAO GRAVADOS UM INDICADOR DE ERRO E AS QUANTIDADES.
07 TRATAR ERROS.
SAO DETECTADOS 3 ERROS: NUMERO DE SIMBOLOS > 154, PRIMEIRA PRODUCAO DA GRAMATICA COM RAIZ EM BRANCO E UMA PRODUCAO FALTANDO ARGUMENTO. OS CONTADORES INDICADOS SAO COLOCADOS EM VALORES ESPECIAIS PARA FINDAR OS CICLOS E E' GRAVADO O NUMERO DO ERRO NA POSICAO 0 DA TABELA DE SIMBOLOS.
NAO TERMINAIS
NUMERO DE SIMBOLOS
NUMERO DE PRODUCOES
ARGUMENTO ACUMULADO
ERRO
NAO TERMINAIS
NUMERO DE SIMBOLOS
PRODUCAO
ARGUMENTO ACUMULADO
NUMERO DE PRODUCOES
NOVA PRODUCAO
TABELA DE SIMBOLOS (ARQTAB)
PONT
NAO TERMINAIS
NUMERO DE SIMBOLOS
POSICAO 0 DA TABELA DE SIMBOLOS

FIGURA IV.5

```
      ****************************************
      *  LISTAGEM FONTE DO PROGRAMA ANAG101  *
      ****************************************


 /*  CODIFICAR SIMBOLOS E PRODUCOES  (1)    -    VERSAO 21/08/78        */ANA0001
0   ANAG101:                                                             ANA0002
    PROC OPTIONS (MAIN);                                                 ANA0003
    DCL  PRODUCAO                          CHAR(80),                     ANA0004
         CARACTER_DA_PRODUCAO(80)          CHAR(1) BASED                 ANA0005
                                           (ADDR(PRODUCAO)),             ANA0006
         REGPROD                           CHAR(80),                     ANA0007
         VETOR_DE_INDICES(40)              BIN FIXED                     ANA0008
                                           BASED(ADDR(REGPROD)),         ANA0009
         BRANCO                            CHAR(1) INIT(' '),            ANA0010
         REGISTRO_BRANCO                   CHAR(80) INIT((80) ' '),      ANA0011
         NOVAPRODUCAO                      BIN FIXED INIT(1),            ANA0012
         MARCA                             BIN FIXED INIT(0),            ANA0013
         UM                                CHAR(1) INIT('1'),            ANA0014
         DOIS                              CHAR(1) INIT('2'),            ANA0015
         TRES                              CHAR(1) INIT('3'),            ANA0016
         RAIZANTERIOR                      BIN FIXED,                    ANA0017
         NAOTERMINAIS                      BIN FIXED INIT(0),            ANA0018
         NUMERO_DE_SIMBOLOS                BIN FIXED,                    ANA0019
         ARGUMENTO_ACUMULADO               BIN FIXED INIT(0),            ANA0020
         GRAVADOS                          BIN FIXED INIT(0),            ANA0021
         SIMBOLO                           CHAR(32) VAR,                 ANA0022
         TAMANHO                           BIN FIXED,                    ANA0023
         PONT                              BIN FIXED,                    ANA0024
         FIM                               BIN FIXED INIT(0),            ANA0025
         NUMERO_DE_PRODUCOES               BIN FIXED INIT(0),            ANA0026
         INDICE_NA_TABELA                  BIN FIXED,                    ANA0027
         TABELA_DE_SIMBOLOS(0:255)         CHAR(32) INIT((32) ' '),      ANA0028
         LIMITE                            BIN FIXED INIT(255),          ANA0029
         K                                 BIN FIXED,                    ANA0030
         BNFONTE                           FILE RECORD INPUT,            ANA0031
         ARQPROD                           FILE RECORD OUTPUT;           ANA0032
0   ON ENDFILE(BNFONTE)                                                  ANA0033
    BEGIN;                                                               ANA0034
         K=1;                                                            ANA0035
         FIM=FIM+1;                                                      ANA0036
    END;                                                                 ANA0037
1   CALL TRATAR_BNF_FONTE;                                               ANA0038
    IF CARACTER_DA_PRODUCAO(1)=BRANCO                                    ANA0039
    THEN CALL TRATAR_ERRO(UM);                                           ANA0040
    DO WHILE(FIM=0);                                                     ANA0041
         PONT=1;                                                         ANA0042
         CALL EXTRAIR_SIMBOLOS_E_BRANCOS;                                ANA0043
         IF TAMANHO>0                                                    ANA0044
         THEN                                                            ANA0045
         DO;                                                             ANA0046
              IF NAOTERMINAIS=LIMITE                                     ANA0047
              THEN CALL TRATAR_ERRO(DOIS);                               ANA0048
              ELSE CALL CODIFICAR_NAOTERMINAIS;                          ANA0049
         END;                                                            ANA0050
         CALL TRATAR_BNF_FONTE;                                          ANA0051
    END;                                                                 ANA0052
    NUMERO_DE_SIMBOLOS=NAOTERMINAIS;                                     ANA0053
    CLOSE FILE(BNFONTE);                                                 ANA0054
    OPEN FILE(BNFONTE);                                                  ANA0055
```

```
0    CALL TRATAR_BNF_FONTE;                                                    ANA0056
     DO WHILE(FIM=1);                                                          ANA0057
          PONT=1;                                                              ANA0058
          NOVAPRODUCAO=1;                                                      ANA0059
          CALL EXTRAIR_SIMBOLOS_E_BRANCOS;                                     ANA0060
          IF TAMANHO>0                                                         ANA0061
          THEN CALL CODIFICAR_TERMINAIS;                                       ANA0062
          ELSE INDICE_NA_TABELA=0;                                             ANA0063
          CALL CODIFICAR_PRODUCOES;                                            ANA0064
          IF PONT=80                                                           ANA0065
          THEN CALL TRATAR_ERRO(TRES);                                         ANA0066
          ELSE                                                                 ANA0067
          DO WHILE(PONT<80);                                                   ANA0068
               CALL EXTRAIR_SIMBOLOS_E_BRANCOS;                                ANA0069
               IF NUMERO_DE_SIMBOLOS=LIMITE                                    ANA0070
               THEN CALL TRATAR_ERRO(DOIS);                                    ANA0071
               ELSE CALL CODIFICAR_TERMINAIS;                                  ANA0072
               CALL CODIFICAR_PRODUCOES;                                       ANA0073
          END;                                                                 ANA0074
          CALL TRATAR_BNF_FONTE;                                               ANA0075
     END;                                                                      ANA0076
0    NOVAPRODUCAO=1;                                                           ANA0077
     INDICE_NA_TABELA=MARCA;                                                   ANA0078
     RAIZANTERIOR=MARCA;                                                       ANA0079
     CALL CODIFICAR_PRODUCOES;                                                 ANA0080
     WRITE FILE(ARQPROD) FROM (REGPROD);                                       ANA0081
     CALL GRAVAR_TABELAS_E_QUANTIDADES;                                        ANA0082
1/*****************************************************************/          ANA0083
 /****************** TRATAR BNF FONTE (1.1) ***********************/          ANA0084
 /*****************************************************************/          ANA0085
     TRATAR_BNF_FONTE:                                                         ANA0086
     PROC;                                                                     ANA0087
     K=0;                                                                      ANA0088
     DO WHILE(K=0);                                                            ANA0089
0         /** LER ARQUIVO DE ENTRADA (1.1.1) **/                              ANA0090
          READ FILE (BNFONTE) INTO (PRODUCAO);                                 ANA0091
0              /** ELIMINAR COMENTARIOS (1.1.2) **/                           ANA0092
          IF CARACTER_DA_PRODUCAO(1)='*'                                       ANA0093
          THEN;                                                                ANA0094
0              /** ELIMINAR REGISTROS BRANCOS (1.1.3) **/                     ANA0095
          ELSE IF PRODUCAO=REGISTRO_BRANCO                                     ANA0096
               THEN;                                                           ANA0097
               ELSE K=1;                                                       ANA0098
     END;                                                                      ANA0099
0    END;    /**** FIM DA TRATAR_BNF_FONTE ****/                              ANA0100
1/*****************************************************************/          ANA0101
 /****************** EXTRAIR SIMBOLOS E BRANCOS (1.2) *************************/ANA0102
 /*****************************************************************/          ANA0103
     EXTRAIR_SIMBOLOS_E_BRANCOS:                                               ANA0104
     PROC;                                                                     ANA0105
0    TAMANHO=0;                                                                ANA0106
     SIMBOLO='';                                                               ANA0107
0         /** EXTRAIR SIMBOLO (1.2.1) **/                                     ANA0108
     IF CARACTER_DA_PRODUCAO(PONT) NE BRANCO                                   ANA0109
     THEN                                                                      ANA0110
     DO K=PONT TO 80;                                                          ANA0111
          IF CARACTER_DA_PRODUCAO(K)=BRANCO                                    ANA0112
          THEN                                                                 ANA0113
          DO;                                                                  ANA0114
               PONT=K;                                                         ANA0115
```

```
                K=80;                                                    ANA0116
           END;                                                          ANA0117
           ELSE                                                          ANA0118
           DO;                                                           ANA0119
                TAMANHO=TAMANHO+1;                                       ANA0120
                IF TAMANHO<33                                            ANA0121
                THEN SIMBOLO=SIMBOLO CAT CARACTER_DA_PRODUCAO(K);        ANA0122
           END;                                                          ANA0123
      END;                                                               ANA0124
0          /** EXTRAIR BRANCOS (1.2.2) **/                             ANA0125
      DO WHILE (CARACTER_DA_PRODUCAO(PONT)=BRANCO);                      ANA0126
           PONT=PONT+1;                                                  ANA0127
           IF PONT=81                                                    ANA0128
           THEN                                                          ANA0129
           DO;                                                           ANA0130
                PONT=80;                                                 ANA0131
                CARACTER_DA_PRODUCAO(80)='*';                            ANA0132
           END;                                                          ANA0133
      END;                                                               ANA0134
0     END;      /******* FIM DE EXTRAIR_SIMBOLOS_E_BRANCOS ********/     ANA0135
1/*****************************************************************/    ANA0136
 /****************** CODIFICAR NAO-TERMINAIS (1.3) *******************/  ANA0137
 /*****************************************************************/    ANA0138
      CODIFICAR_NAOTERMINAIS:                                            ANA0139
      PROC;                                                              ANA0140
0          /** VERIFICAR SE NAO-TERMINAL JA' FOI GRAVADO (1.3.1) **/    ANA0141
      INDICE_NA_TABELA=NAOTERMINAIS+1;                                   ANA0142
      DO K=1 TO NAOTERMINAIS;                                            ANA0143
           IF TABELA_DE_SIMBOLOS(K)=SIMBOLO                              ANA0144
           THEN                                                          ANA0145
           DO;                                                           ANA0146
                INDICE_NA_TABELA=0;                                      ANA0147
                K=NAOTERMINAIS;                                          ANA0148
           END;                                                          ANA0149
      END;                                                               ANA0150
      IF INDICE_NA_TABELA>0                                              ANA0151
      THEN                                                               ANA0152
      DO;                                                                ANA0153
0          /** CONTAR NAO TERMINAIS (1.3.2) **/                        ANA0154
           NAOTERMINAIS=NAOTERMINAIS+1;                                  ANA0155
0          /** TABELAR NAO-TERMINAL FONTE (1.3.3) **/                  ANA0156
           TABELA_DE_SIMBOLOS(NAOTERMINAIS)=SIMBOLO;                     ANA0157
      END;                                                               ANA0158
0     END;      /***** FIM DE CODIFICAR_NAOTERMINAIS ******/            ANA0159
1/*****************************************************************/    ANA0160
 /******************** CODIFICAR TERMINAIS (1.4) **********************/  ANA0161
 /*****************************************************************/    ANA0162
      CODIFICAR_TERMINAIS:                                               ANA0163
      PROC;                                                              ANA0164
0          /** CONSULTAR TABELA DE SIMBOLOS FONTE (1.4.1) **/           ANA0165
      INDICE_NA_TABELA=0;                                                ANA0166
      DO K=1 TO NUMERO_DE_SIMBOLOS;                                      ANA0167
           IF TABELA_DE_SIMBOLOS(K)=SIMBOLO                              ANA0168
           THEN                                                          ANA0169
           DO;                                                           ANA0170
                INDICE_NA_TABELA=K;                                      ANA0171
                K=NUMERO_DE_SIMBOLOS;                                    ANA0172
           END;                                                          ANA0173
      END;                                                               ANA0174
      IF INDICE_NA_TABELA=0                                              ANA0175
      THEN                                                               ANA0176
      DO;                                                                ANA0177
```

```
0         /** CONTAR SIMBOLOS TABELADOS (1.4.2) **/                          ANA0178
          NUMERO_DE_SIMBOLOS=NUMERO_DE_SIMBOLOS+1;                           ANA0179
          INDICE_NA_TABELA=NUMERO_DE_SIMBOLOS;                               ANA0180
0         /** TABELAR TERMINAL FONTE (1.4.3) **/                             ANA0181
          TABELA_DE_SIMBOLOS(NUMERO_DE_SIMBOLOS)=SIMBOLO;                    ANA0182
     END;                                                                    ANA0183
0    END;      /****** FIM DE CODIFICAR_TERMINAIS *****/                      ANA0184
1/*************************************************************************/ ANA0185
 /********************* CODIFICAR PRODUCOES (1.5) ************************/ ANA0186
 /*************************************************************************/ ANA0187
     CODIFICAR_PRODUCOES:                                                    ANA0188
     PROC;                                                                   ANA0189
     IF INDICE_NA_TABELA=0                                                   ANA0190
0         /** FAZER INDICE IGUAL AO INDICE DA RAIZ ANTERIOR (1.5.1) **/     ANA0191
     THEN INDICE_NA_TABELA=RAIZANTERIOR;                                     ANA0192
     IF NOVAPRODUCAO=1                                                       ANA0193
     THEN                                                                    ANA0194
     DO;                                                                     ANA0195
0         /** MARCAR FIM DE PRODUCAO (1.5.2) **/                            ANA0196
               /** GRAVAR MARCA DE FIM DE PRODUCAO (1.5.2.1) **/            ANA0197
          CALL GRAVAR_VETOR_DE_INDICES(MARCA);                               ANA0198
               /** GRAVAR TAMANHO ACUMULADO DO ARGUMENTO (1.5.2.2) **/      ANA0199
          CALL GRAVAR_VETOR_DE_INDICES(ARGUMENTO_ACUMULADO);                 ANA0200
0         /** CONTAR PRODUCOES (1.5.3) **/                                   ANA0201
          NUMERO_DE_PRODUCOES=NUMERO_DE_PRODUCOES+1;                         ANA0202
0         /** GUARDAR RAIZ ANTERIOR (1.5.4) **/                              ANA0203
          RAIZANTERIOR=INDICE_NA_TABELA;                                     ANA0204
0         NOVAPRODUCAO=0;                                                    ANA0205
          ARGUMENTO_ACUMULADO=ARGUMENTO_ACUMULADO-1;                         ANA0206
     END;                                                                    ANA0207
0         /** GRAVAR INDICE (1.5.5) **/                                      ANA0208
     CALL GRAVAR_VETOR_DE_INDICES(INDICE_NA_TABELA);                         ANA0209
0         /** ACUMULAR TAMANHO DO ARGUMENTO (1.5.6) **/                      ANA0210
     ARGUMENTO_ACUMULADO=ARGUMENTO_ACUMULADO+1;                              ANA0211
1/*************************************************************************/ ANA0212
 /****************** GRAVAR VETOR DE INDICES (1.5.5) ********************/ ANA0213
 /*************************************************************************/ ANA0214
     GRAVAR_VETOR_DE_INDICES:                                                ANA0215
     PROC(INDICE_NA_TABELA);                                                 ANA0216
     DCL INDICE_NA_TABELA                              BIN FIXED;            ANA0217
0    GRAVADOS=GRAVADOS+1;                                                    ANA0218
     IF GRAVADOS>40                                                          ANA0219
     THEN                                                                    ANA0220
     DO;                                                                     ANA0221
          WRITE FILE(ARQPROD) FROM (REGPROD);                                ANA0222
          GRAVADOS=1;                                                        ANA0223
     END;                                                                    ANA0224
     VETOR_DE_INDICES(GRAVADOS)=INDICE_NA_TABELA;                            ANA0225
0    END;      /******* FIM DE GRAVAR_VETOR_DE_INDICES *******/              ANA0226
0    END;    /***** FIM DE CODIFICAR_PRODUCOES ******/                       ANA0227
1/*************************************************************************/ANA0228
 /****************** GRAVAR TABELAS E QUANTIDADES (1.6) ******************/ANA0229
 /*************************************************************************/ANA0230
     GRAVAR_TABELAS_E_QUANTIDADES:                                           ANA0231
     PROC;                                                                   ANA0232
     DCL 1 TABELA_GRAVADA                    BASED(ADDR(REGPROD)),           ANA0233
           2 SIMBOLO_GRAVADO(2)              CHAR(32),                       ANA0234
           2 XXX                             CHAR(16),                       ANA0235
         ARQTAB                              FILE RECORD OUTPUT;             ANA0236
```

```
0         /** GRAVAR QUANTIDADES (1.6.1) **/                          ANA0237
     GRAVADOS=1;                                                       ANA0238
     IF TABELA_DE_SIMBOLOS(0)=BRANCO                                   ANA0239
     THEN                                                              ANA0240
     DO;                                                               ANA0241
          VETOR_DE_INDICES(1)=0;                                       ANA0242
          VETOR_DE_INDICES(2)=NAOTERMINAIS;                            ANA0243
          VETOR_DE_INDICES(3)=NUMERO_DE_SIMBOLOS;                      ANA0244
          VETOR_DE_INDICES(4)=NUMERO_DE_PRODUCOES-1;                   ANA0245
          VETOR_DE_INDICES(5)=ARGUMENTO_ACUMULADO;                     ANA0246
     END;                                                              ANA0247
     ELSE SIMBOLO_GRAVADO(1)=TABELA_DE_SIMBOLOS(0);                    ANA0248
0         /** GRAVAR TABELAS DE SIMBOLOS (1.6.2) **/                  ANA0249
     DO K=1 TO NUMERO_DE_SIMBOLOS;                                     ANA0250
          GRAVADOS=GRAVADOS+1;                                         ANA0251
          IF GRAVADOS>2                                                ANA0252
          THEN                                                         ANA0253
          DO;                                                          ANA0254
               WRITE FILE (ARQTAB) FROM (REGPROD);                     ANA0255
               GRAVADOS=1;                                             ANA0256
          END;                                                         ANA0257
          SIMBOLO_GRAVADO(GRAVADOS)=TABELA_DE_SIMBOLOS(K);             ANA0258
     END;                                                              ANA0259
     WRITE FILE(ARQTAB) FROM (REGPROD);                                ANA0260
0    END;      /****** FIM DE GRAVAR_TABELAS_E_QUANTIDADES *******/     ANA0261
1/*****************************************************************/  ANA0262
 /********************* TRATAR ERROS (1.7) ***************************/  ANA0263
 /*****************************************************************/  ANA0264
     TRATAR_ERRO:                                                      ANA0265
     PROC(ERRO);                                                       ANA0266
     DCL ERRO                                  CHAR(1);                ANA0267
0    PONT=80;                                                          ANA0268
     FIM=80;                                                           ANA0269
     NAOTERMINAIS=1;                                                   ANA0270
     NUMERO_DE_SIMBOLOS=1;                                             ANA0271
     TABELA_DE_SIMBOLOS(0)=ERRO;                                       ANA0272
0    END;      /******* FIM DE TRATAR_ERROS ********/                  ANA0273
0    END;       /***** FIM DE ANAG101 *****/                           ANA0274
```

ANAG102 - GERAR MATRIZES PARCIAIS (2).

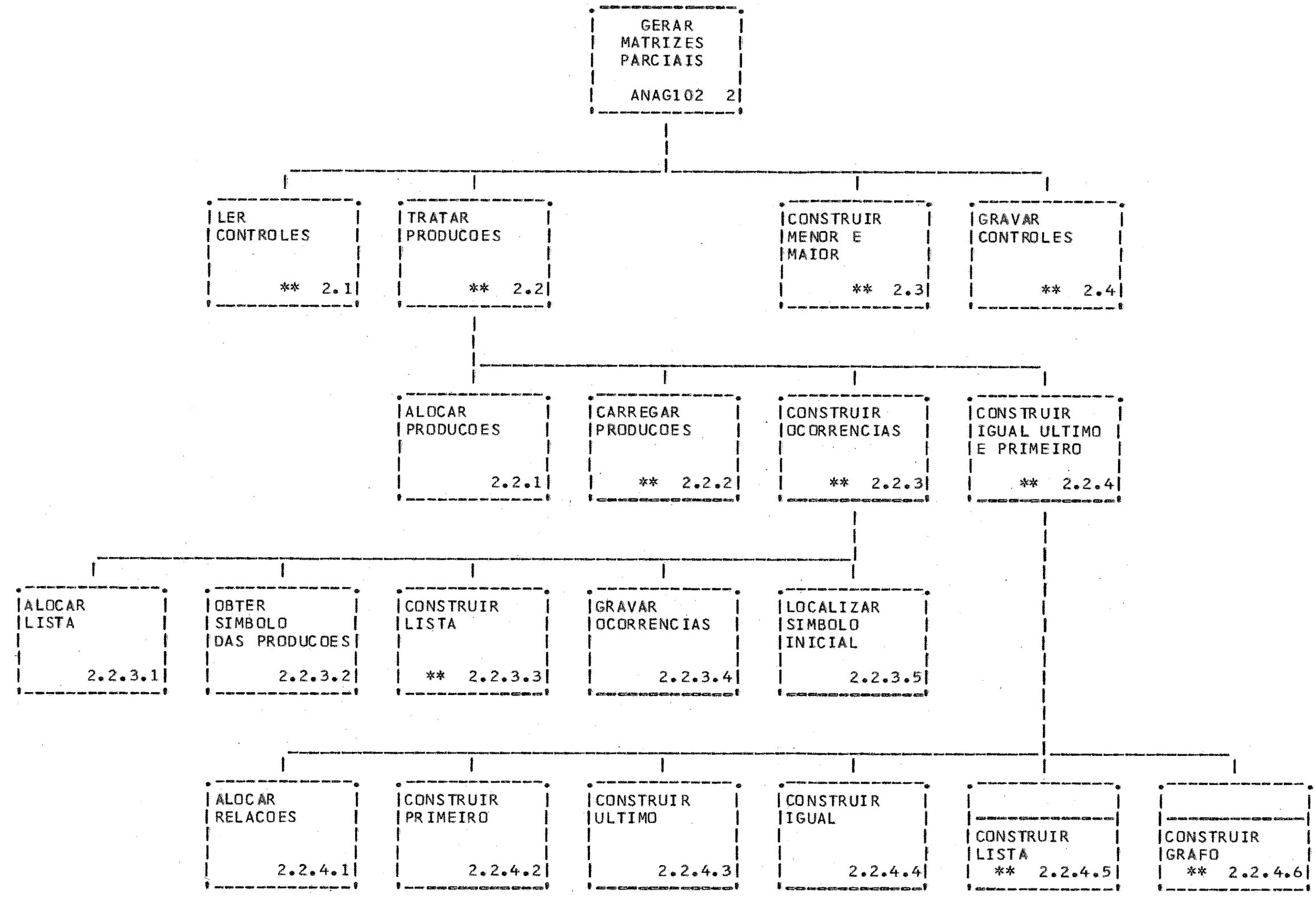


FIGURA IV.6

ANAG102 - CONSTRUIR MENOR E MAIOR (2.3).

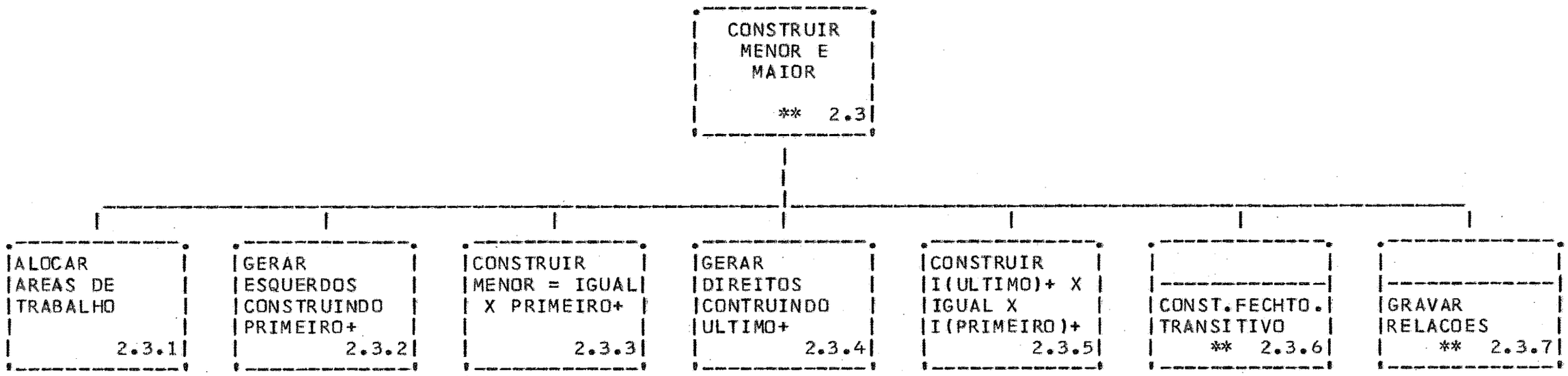


FIGURA IV.7

ANAG102 - GERAR MATRIZES PARCIAIS (2)

ENTRADA

TABELA DE SIMBOLOS (ARQTAB)

PRODUCOES CODIFICADAS (ARQPROD)

ERRO

PROCESSO

01 LER CONTROLES.

PRIMEIRO REGISTRO DE ARQTAB E' LIDO E SAO EXTRAIDOS OS CONTROLES INDICADOS. UM DELES E' A INDICACAO DE ERRO VINDO DE ANAG101

02 TRATAR PRODUCOES.

SAIDA

ERRO

NAO TERMINAIS

NUMERO DE PRODUCOE

TAMANHO DOS ARG UMENTOS

NUMERO DE SIMBOLOS

OCORRENCIAS DOS SIMBOLOS (ARQOCOR)

ERRO

RELACAO PRIMEIRO

RELACAO ULTIMO

RELACAO IGUAL

SIMBOLO INICIAL

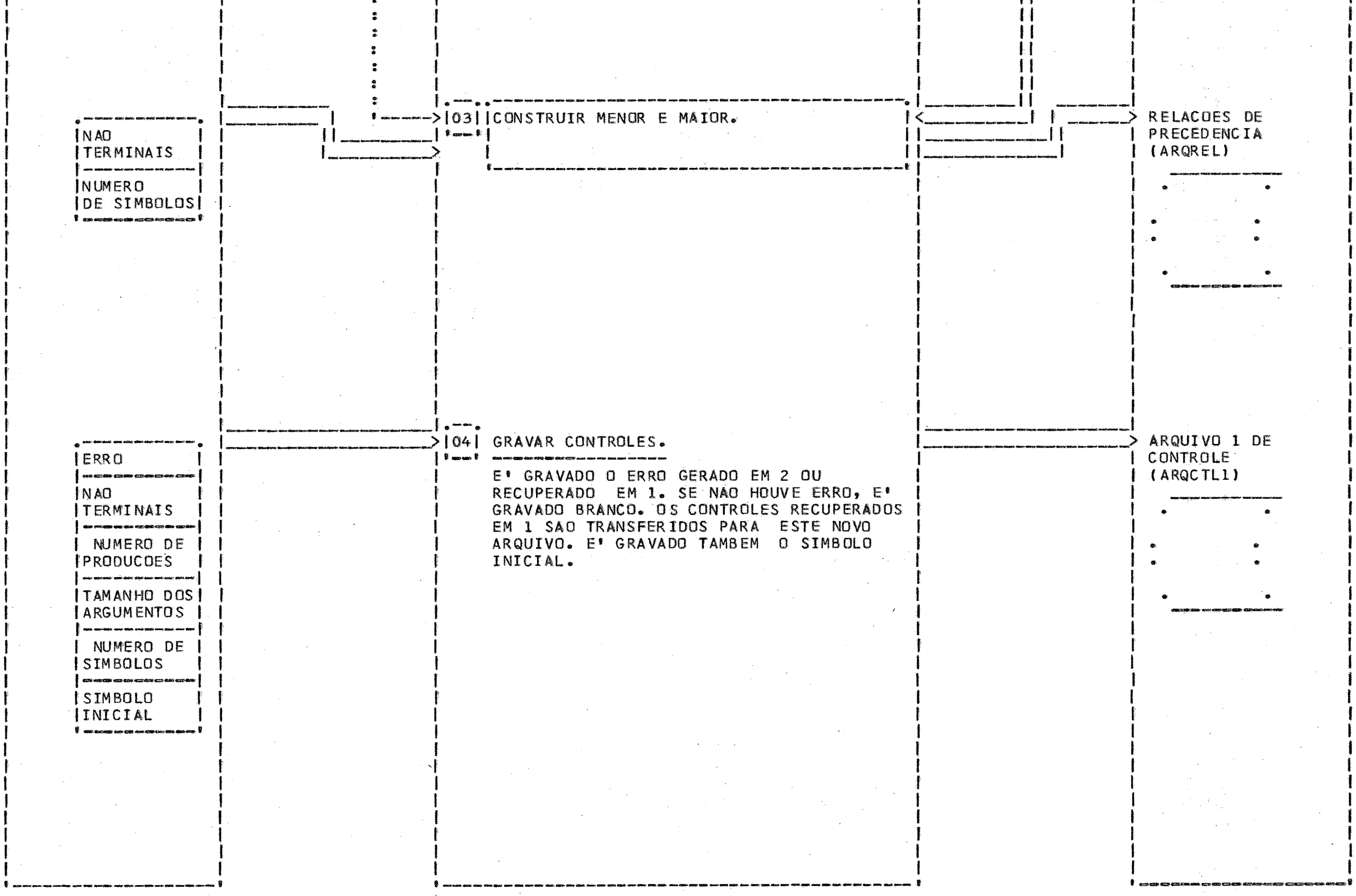

NAO TERMINAIS
NUMERO DE SIMBOLOS
03
CONSTRUIR MENOR E MAIOR.
RELACOES DE PRECEDENCIA (ARQREL)
ERRO
NAO TERMINAIS
NUMERO DE PRODUCOES
TAMANHO DOS ARGUMENTOS
NUMERO DE SIMBOLOS
SIMBOLO INICIAL
04
GRAVAR CONTROLES.
E' GRAVADO O ERRO GERADO EM 2 OU RECUPERADO EM 1. SE NAO HOUVE ERRO, E' GRAVADO BRANCO. OS CONTROLES RECUPERADOS EM 1 SAO TRANSFERIDOS PARA ESTE NOVO ARQUIVO. E' GRAVADO TAMBEM O SIMBOLO INICIAL.
ARQUIVO 1 DE CONTROLE (ARQCTL1)

FIGURA IV.8

TRATAR PRODUCOES (2.2)

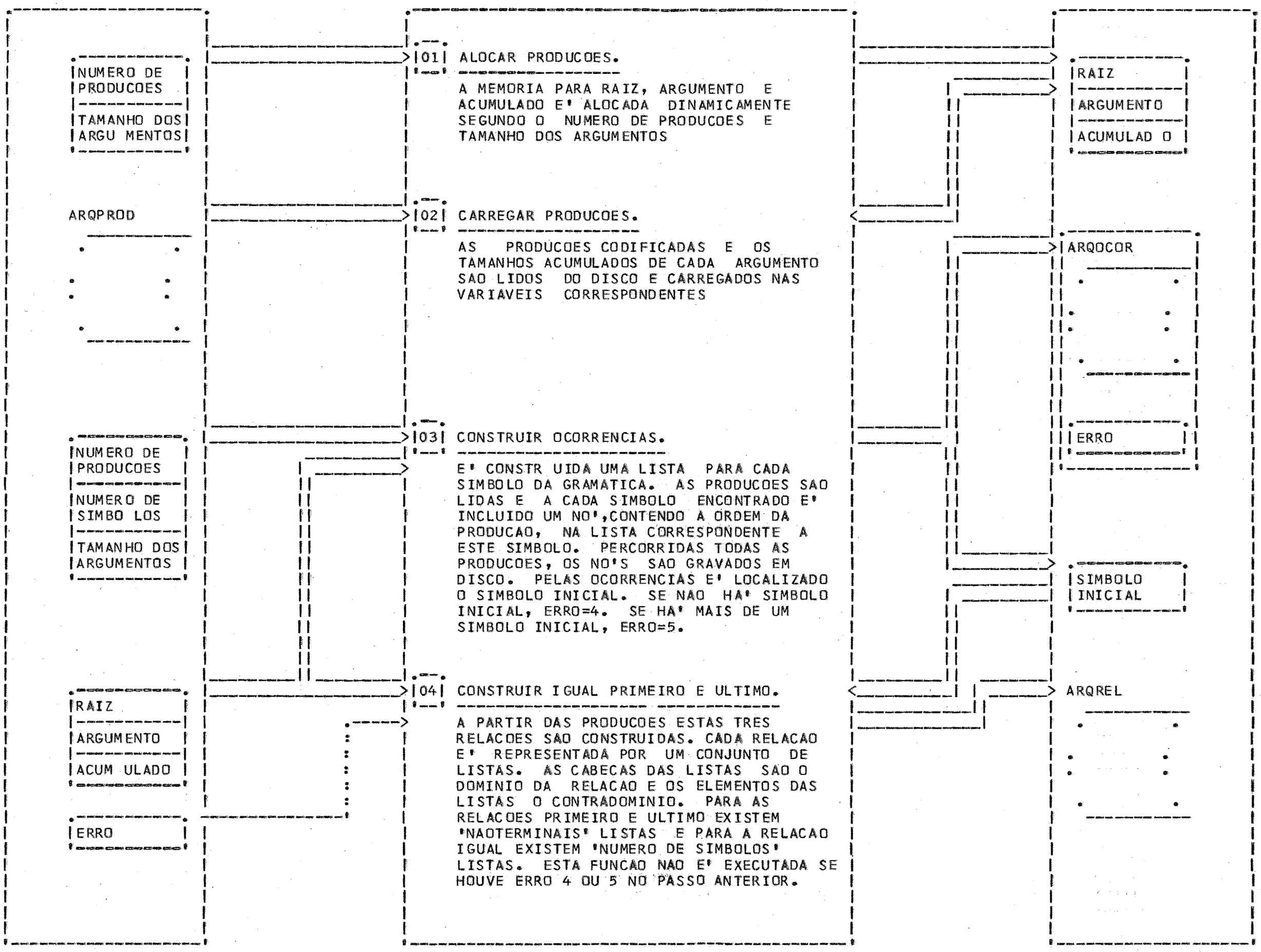


FIGURA IV.9

CONSTRUIR MENOR E MAIOR (2.3)

ENTRADA

- NUMERO DE SIMBOLOS
- RELACAO PRIMEIRO
- RELACAO IGUAL
- RELACAO ULTIMO
- ESQUERDOS
- RELACAO R
- SIMBOLO S
- PAR PRECEDENCIA

PROCESSO

01 ALOCAR AREAS DE TRABALHO.

02 GERAR ESQUERDOS CONSTRUINDO PRIMEIRO+ .

UMA LISTA DE IGUAL E' OBTIDA. PARA CADA SIMBOLO DA LISTA SAO OBTIDOS OS SIMBOLOS COM OS QUAIS ELE SE RELACIONA POR PRIM EIRO+ . A UNIAO DOS SIMBOLOS OBTIDOS PARA TODOS ELEMENTOS DA LISTA GERA ESQUERDOS .

03 CONSTRUIR MENOR = IGUAL X PRIMEIRO+ .

O PRODUTO DA CABECA DE LISTA COM ESQUERDOS GERA UM SUBCONJUNTO DE MENOR ONDE CADA ELEMENTO E' UM PAR/PRECEDENCIA.

04 GERAR DIREITOS CONSTRUINDO ULTIMO+ .

PARA A CABECA DA LISTA SAO OBTIDOS OS SIMBOLOS COM OS ELA SE RELACIONA POR ULTIMO+, GERANDO DIREITOS.

05 CONSTRUIR MAIOR = I(ULTIMO)+ X IGUAL X I(ULTIMO)+ .

A ESQUERDOS SAO ACRESC ENTADOS OS ELEMENTOS DA LISTA. O PRODUTO ESQUERDOS X DIREITOS E' UM SUBCONJUNTO DE MAIOR ONDE CADA ELEMENTO E' UM PAR/PRECEDENCIA

06 CONSTRUIR FECHAMENTO TRANSITIVO .

DADA UMA RELACAO R, UM SIMBOLO S E UM CONJUNTO EXTREMOS, SAO OBTIDOS OS SIMBOLOS COM OS QUAIS S SE RELACIONA POR R+ . ESTES SIMBOLOS SAO ACRESCENTADOS A EXTREMOS. R PODE SER PRIMEIRO E ULTIMO E EXTREMOS PODE SER ESQUERDOS E DIREITOS.

07 GRAVAR RELACOES .

UM PAR/PRECEDENCIA E' GRAVADO.

SAIDA

- DIREITOS
- ESQUERDOS
- PAR PRECEDENCIA
- ESQUERDOS
- PAR PRECEDENCIA
- DIREITOS
- PAR PRECEDENCIA
- EXTREMOS
- ARQREL

FIGURA IV.10

```
      ****************************************
      *  LISTAGEM FONTE DO PROGRAMA ANAG102  *
      ****************************************

 /* GERAR MATRIZES PARCIAIS  (2)  (VERSAO 22/11/78)                    */ANA0001
    ANAG102:                                                             ANA0002
    PROC OPTIONS(MAIN);                                                  ANA0003
0   DCL NAOTERMINAIS                        BIN FIXED,                   ANA0004
        NUMERO_DE_SIMBOLOS                  BIN FIXED,                   ANA0005
        NUMERO_DE_PRODUCOES                 BIN FIXED,                   ANA0006
        TAMANHO_DOS_ARGUMENTOS              BIN FIXED,                   ANA0007
        SIMBOLO_INICIAL                     BIN FIXED INIT(0),           ANA0008
        DOMINIO_IGUAL(0:1)                  BIN FIXED CONTROLLED,        ANA0009
        PONTEIRO_IGUAL(1)                   BIN FIXED CONTROLLED,        ANA0010
        CONTRADOMINIO_IGUAL(1)              BIN FIXED CONTROLLED,        ANA0011
        DOMINIO_PRIMEIRO(0:1)               BIN FIXED CONTROLLED,        ANA0012
        CONTRADOMINIO_PRIMEIRO(1)           BIN FIXED CONTROLLED,        ANA0013
        DOMINIO_ULTIMO(0:1)                 BIN FIXED CONTROLLED,        ANA0014
        CONTRADOMINIO_ULTIMO(1)             BIN FIXED CONTROLLED,        ANA0015
        ERRO                                CHAR(1) INIT(' ');           ANA0016
0   /** VARIAVEIS GLOBAIS DE TRABALHO **/                                ANA0017
    DCL BRANCO                              CHAR(1) INIT(' '),           ANA0018
        (I,J,K,N,X,Y,S,T)                   BIN FIXED;                   ANA0019
-   CALL LER_CONTROLES;                                                  ANA0020
    IF ERRO = BRANCO                                                     ANA0021
    THEN                                                                 ANA0022
    DO;                                                                  ANA0023
         CALL TRATAR_PRODUCOES;                                          ANA0024
         IF ERRO = BRANCO                                                ANA0025
         THEN CALL CONSTRUIR_MENOR_MAIOR;                                ANA0026
         ELSE;                                                           ANA0027
    END;                                                                 ANA0028
    ELSE;                                                                ANA0029
    CALL GRAVAR_CONTROLES(ERRO);                                         ANA0030
1/*************************************************************************/ANA0031
 /******************** LER CONTROLES (2.1) ***************************/ ANA0032
 /*************************************************************************/ANA0033
    LER_CONTROLES:                                                       ANA0034
    PROC;                                                                ANA0035
0   DCL P                                   POINTER,                     ANA0036
        CONTROLES(40)                       BIN FIXED BASED(P),          ANA0037
        CARATER(80)                         CHAR(1) BASED(P),            ANA0038
        ARQTAB                              FILE RECORD INPUT;           ANA0039
0   READ FILE (ARQTAB) SET (P);                                          ANA0040
    IF CONTROLES(1)=0                                                    ANA0041
    THEN;                                                                ANA0042
    ELSE ERRO=CARATER(1);                                                ANA0043
0   NAOTERMINAIS=CONTROLES(2);                                           ANA0044
    NUMERO_DE_SIMBOLOS=CONTROLES(3);                                     ANA0045
    NUMERO_DE_PRODUCOES=CONTROLES(4);                                    ANA0046
    TAMANHO_DOS_ARGUMENTOS=CONTROLES(5);                                 ANA0047
0   CLOSE FILE (ARQTAB);                                                 ANA0048
0   END;   /*** FIM DE LER_CONTROLES ***/                               ANA0049
1/*************************************************************************/ANA0050
 /******************** TRATAR PRODUCOES (2.2) ********************/      ANA0051
 /*************************************************************************/ANA0052
    TRATAR_PRODUCOES:                                                    ANA0053
    PROC;                                                                ANA0054
    DCL RAIZ(1)                             BIN FIXED CONTROLLED,        ANA0055
```

```
        ARGUMENTO(1)                          BIN FIXED CONTROLLED,      ANA0056
        ACUMULADO(0:1)                        BIN FIXED CONTROLLED;      ANA0057
0   /*** ALOCAR PRODUCOES (2.2.1) ***/                                   ANA0058
    ALLOCATE RAIZ(NUMERO_DE_PRODUCOES);                                  ANA0059
    ALLOCATE ARGUMENTO(TAMANHO_DOS_ARGUMENTOS);                          ANA0060
    ALLOCATE ACUMULADO(0:NUMERO_DE_PRODUCOES) INIT(0);                   ANA0061
0   CALL CARREGAR_PRODUCOES;                                             ANA0062
    CALL CONSTRUIR_OCORRENCIAS;                                          ANA0063
    IF ERRO NE BRANCO                                                    ANA0064
    THEN;                                                                ANA0065
    ELSE CALL CONSTRUIR_IGUAL_PRIMEIRO_ULTIMO;                           ANA0066
1/*****************************************************************/    ANA0067
 /*************** CARREGAR PRODUCOES (2.2.2) ********************/     ANA0068
 /*****************************************************************/    ANA0069
    CARREGAR_PRODUCOES:                                                  ANA0070
    PROC;                                                                ANA0071
    DCL P                                 POINTER,                       ANA0072
        VETORPROD(40)                     BIN FIXED BASED(P),            ANA0073
        ARQPROD                           FILE RECORD INPUT;             ANA0074
0   X=40;                                                                ANA0075
    Y,S,T=0;                                                             ANA0076
    DO K=1 TO 3;                                                         ANA0077
         CALL LER_ARQUIVO_CODIGOS;                                       ANA0078
    END;                                                                 ANA0079
0   DO WHILE (VETORPROD(X)>0);                                           ANA0080
         Y=Y+1;                                                          ANA0081
         RAIZ(Y)=VETORPROD(X);                                           ANA0082
         CALL LER_ARQUIVO_CODIGOS;                                       ANA0083
         DO WHILE (VETORPROD(X)>0);                                      ANA0084
              S=S+1;                                                     ANA0085
              ARGUMENTO(S)=VETORPROD(X);                                 ANA0086
              CALL LER_ARQUIVO_CODIGOS;                                  ANA0087
         END;                                                            ANA0088
         CALL LER_ARQUIVO_CODIGOS;                                       ANA0089
         T=T+1;                                                          ANA0090
         ACUMULADO(T)=VETORPROD(X);                                      ANA0091
         CALL LER_ARQUIVO_CODIGOS;                                       ANA0092
    END;                                                                 ANA0093
    CLOSE FILE (ARQPROD);                                                ANA0094
1/*****************************************************************/    ANA0095
 /*************** LER ARQUIVO DE CODIGOS (2.2.2.1) ***************/     ANA0096
 /*****************************************************************/    ANA0097
    LER_ARQUIVO_CODIGOS:                                                 ANA0098
    PROC;                                                                ANA0099
0   X=X+1;                                                               ANA0100
    IF X>40                                                              ANA0101
    THEN                                                                 ANA0102
    DO;                                                                  ANA0103
         READ FILE (ARQPROD) SET (P);                                    ANA0104
         X=1;                                                            ANA0105
    END;                                                                 ANA0106
0   END;    /*** FIM DE LER_ARQUIVO_CODIGOS ***/                        ANA0107
0   END;    /*** FIM DE CARREGAR_PRODUCOES ***/                         ANA0108
1/*****************************************************************/    ANA0109
 /*************** CONSTRUIR OCORRENCIAS (2.2.3) ******************/     ANA0110
 /*****************************************************************/    ANA0111
    CONSTRUIR_OCORRENCIAS:                                               ANA0112
    PROC;                                                                ANA0113
    DCL REGISTRO                              CHAR(80),                  ANA0114
        P                                     POINTER,                   ANA0115
```

```
      VETOR(40)                          BIN FIXED BASED(P),          ANA0116
      QUATRO                             CHAR(1) INIT('4'),           ANA0117
      CINCO                              CHAR(1) INIT('5'),           ANA0118
      ARQOCOR                            FILE RECORD OUTPUT;          ANA0119
0     DCL OCORRENCIA(1)                  BIN FIXED CONTROLLED,        ANA0120
      PONTEIRO(1)                        BIN FIXED CONTROLLED,        ANA0121
      CABECA_DE_LISTA(1)                 BIN FIXED CONTROLLED,        ANA0122
      DISPONIVEL                         BIN FIXED INIT(1),           ANA0123
      INDICADOR(256)                     BIT(1) INIT((256)('0'B));    ANA0124
0     /*** ALOCAR LISTA (2.2.3.1) ***/                                ANA0125
      J = NUMERO_DE_PRODUCOES + TAMANHO_DOS_ARGUMENTOS;              ANA0126
      ALLOCATE OCORRENCIA(J);                                        ANA0127
      ALLOCATE PONTEIRO(J);                                          ANA0128
      ALLOCATE CABECA_DE_LISTA(NUMERO_DE_SIMBOLOS);                  ANA0129
      DO J=1 TO NUMERO_DE_SIMBOLOS;                                  ANA0130
           CABECA_DE_LISTA(J)=0;                                     ANA0131
      END;                                                           ANA0132
0     /*** OBTER SIMBOLO DAS PRODUCOES (2.2.3.2) ***/                ANA0133
      DO J=1 TO NUMERO_DE_PRODUCOES;                                 ANA0134
           S=RAIZ(J);                                                ANA0135
           CALL CONSTRUIR_LISTA;                                     ANA0136
           X=ACUMULADO(J-1)+1;                                       ANA0137
           DO K=X TO ACUMULADO(J);                                   ANA0138
                S=ARGUMENTO(K);                                      ANA0139
                INDICADOR(S)='1'B;                                   ANA0140
                CALL CONSTRUIR_LISTA;                                ANA0141
           END;                                                      ANA0142
      END;                                                           ANA0143
1     /*** GRAVAR OCORRENCIAS (2.2.3.4) ***/                         ANA0144
      X=0;                                                           ANA0145
      P=ADDR(REGISTRO);                                              ANA0146
      DO S=1 TO NUMERO_DE_SIMBOLOS;                                  ANA0147
           T=CABECA_DE_LISTA(S);                                     ANA0148
           DO WHILE (T>0);                                           ANA0149
                CALL GRAVAR_ARQUIVO_DE_OCORRENCIAS;                  ANA0150
                VETOR(X)=OCORRENCIA(T);                              ANA0151
                T=PONTEIRO(T);                                       ANA0152
           END;                                                      ANA0153
           CALL GRAVAR_ARQUIVO_DE_OCORRENCIAS;                       ANA0154
           VETOR(X)=0;                                               ANA0155
      END;                                                           ANA0156
      X=40;                                                          ANA0157
      CALL GRAVAR_ARQUIVO_DE_OCORRENCIAS;                            ANA0158
0     /*** LOCALIZAR SIMBOLO INICIAL (2.2.3.5) ***/                  ANA0159
      J=0;                                                           ANA0160
      DO K=1 TO NUMERO_DE_SIMBOLOS;                                  ANA0161
           IF INDICADOR(K)='0'B                                      ANA0162
           THEN                                                      ANA0163
           DO;                                                       ANA0164
                SIMBOLO_INICIAL=K;                                   ANA0165
                J=J+1;                                               ANA0166
           END;                                                      ANA0167
      END;                                                           ANA0168
      IF SIMBOLO_INICIAL=0                                           ANA0169
      THEN ERRO=QUATRO;                                              ANA0170
      ELSE IF J>1                                                    ANA0171
           THEN ERRO=CINCO;                                          ANA0172
           ELSE;                                                     ANA0173
      FREE OCORRENCIA;                                               ANA0174
      FREE PONTEIRO;                                                 ANA0175
```

```
    FREE CABECA_DE_LISTA;                                                    ANA0176
    CLOSE FILE (ARQOCOR);                                                    ANA0177
1/*****************************************************************/        ANA0178
 /****************** CONSTRUIR A LISTA (2.2.3.3) ******************/        ANA0179
 /*****************************************************************/        ANA0180
    CONSTRUIR_LISTA:                                                         ANA0181
    PROC;                                                                    ANA0182
    IF CABECA_DE_LISTA(S)=0                                                  ANA0183
    THEN CABECA_DE_LISTA(S)=DISPONIVEL;                                      ANA0184
    ELSE                                                                     ANA0185
    DO;                                                                      ANA0186
         T=CABECA_DE_LISTA(S);                                               ANA0187
         DO WHILE (T>0);                                                     ANA0188
              I=T;                                                           ANA0189
              T=PONTEIRO(T);                                                 ANA0190
         END;                                                                ANA0191
         PONTEIRO(I)=DISPONIVEL;                                             ANA0192
    END;                                                                     ANA0193
    OCORRENCIA(DISPONIVEL)=J;                                                ANA0194
    PONTEIRO(DISPONIVEL)=0;                                                  ANA0195
    DISPONIVEL=DISPONIVEL+1;                                                 ANA0196
0   END;   /*** FIM DE CONSTRUIR_LISTA ***/                                 ANA0197
1/*****************************************************************/        ANA0198
 /*********** GRAVAR ARQUIVO DE OCORRENCIAS (2.2.3.4.1) ***********/        ANA0199
 /*****************************************************************/        ANA0200
    GRAVAR_ARQUIVO_DE_OCORRENCIAS:                                           ANA0201
    PROC;                                                                    ANA0202
    X=X+1;                                                                   ANA0203
    IF X>40                                                                  ANA0204
    THEN                                                                     ANA0205
    DO;                                                                      ANA0206
         WRITE FILE (ARQOCOR) FROM (REGISTRO);                               ANA0207
         X=1;                                                                ANA0208
    END;                                                                     ANA0209
0   END;    /*** FIM DE GRAVAR_ARQUIVO_DE_OCORRENCIAS ***/                  ANA0210
0   END;   /*** FIM DE CONSTRUIR_OCORRENCIAS ***/                          ANA0211
1/*****************************************************************/        ANA0212
 /*********** CONSTRUIR IGUAL PRIMEIRO E ULTIMO (2.2.4) ***********/        ANA0213
 /*****************************************************************/        ANA0214
    CONSTRUIR_IGUAL_PRIMEIRO_ULTIMO:                                         ANA0215
    PROC;                                                                    ANA0216
    DCL DISPONIVEL                          BIN FIXED INIT(1),               ANA0217
        P                                   POINTER,                         ANA0218
        CABECA_DE_LISTA(0:255)              BIN FIXED BASED(P);              ANA0219
-        /*** ALOCAR RELACOES (2.2.4.1) ***/                                 ANA0220
0   NUMERO_DE_SIMBOLOS=NUMERO_DE_SIMBOLOS+1;                                 ANA0221
    I = TAMANHO_DOS_ARGUMENTOS - NUMERO_DE_PRODUCOES + 2;                    ANA0222
    ALLOCATE DOMINIO_IGUAL(0:NUMERO_DE_SIMBOLOS);                            ANA0223
    ALLOCATE DOMINIO_PRIMEIRO (0:NAOTERMINAIS);                              ANA0224
    ALLOCATE DOMINIO_ULTIMO (0:NAOTERMINAIS);                                ANA0225
    IF NUMERO_DE_PRODUCOES > I                                               ANA0226
    THEN J=NUMERO_DE_PRODUCOES;                                              ANA0227
    ELSE J=I;                                                                ANA0228
    ALLOCATE CONTRADOMINIO_IGUAL (J);                                        ANA0229
    ALLOCATE PONTEIRO_IGUAL(J);                                              ANA0230
    ALLOCATE CONTRADOMINIO_PRIMEIRO (NUMERO_DE_PRODUCOES) ;                  ANA0231
    ALLOCATE CONTRADOMINIO_ULTIMO (NUMERO_DE_PRODUCOES);                     ANA0232
1        /*** CONSTRUIR PRIMEIRO (2.2.4.2) ***/                             ANA0233
-   P=ADDR(DOMINIO_PRIMEIRO);                                                ANA0234
    DISPONIVEL = 1;                                                          ANA0235
```

```
0    DO J=0 TO NAOTERMINAIS;                                                    ANA0236
          DOMINIO_PRIMEIRO(J)=0;                                                ANA0237
     END;                                                                       ANA0238
0    DO J=1 TO NUMERO_DE_PRODUCOES;                                             ANA0239
          S=RAIZ(J);                                                            ANA0240
          N=ARGUMENTO(ACUMULADO(J-1)+1);                                        ANA0241
          CALL CONSTRUIR_LISTA;                                                 ANA0242
     END;                                                                       ANA0243
0    CALL CONSTRUIR_GRAFO ( CONTRADOMINIO_PRIMEIRO );                           ANA0244
1         /*** CONSTRUIR ULTIMO (2.2.4.3) ***/                                 ANA0245
-    P=ADDR(DOMINIO_ULTIMO);                                                    ANA0246
     DISPONIVEL=1;                                                              ANA0247
0    DO J=0 TO NAOTERMINAIS;                                                    ANA0248
          DOMINIO_ULTIMO(J)=0;                                                  ANA0249
     END;                                                                       ANA0250
0    DO J=1 TO NUMERO_DE_PRODUCOES;                                             ANA0251
          S=RAIZ(J);                                                            ANA0252
          N=ARGUMENTO(ACUMULADO(J));                                            ANA0253
          CALL CONSTRUIR_LISTA;                                                 ANA0254
     END;                                                                       ANA0255
0    CALL CONSTRUIR_GRAFO (CONTRADOMINIO_ULTIMO );                              ANA0256
1         /*** CONSTRUIR IGUAL (2.2.4.4) ***/                                  ANA0257
-    P=ADDR(DOMINIO_IGUAL);                                                     ANA0258
     DISPONIVEL=1;                                                              ANA0259
0    DO J=0 TO NUMERO_DE_SIMBOLOS;                                              ANA0260
          DOMINIO_IGUAL(J)=0;                                                   ANA0261
     END;                                                                       ANA0262
0    DO J=1 TO NUMERO_DE_PRODUCOES;                                             ANA0263
          X=ACUMULADO(J-1)+1;                                                   ANA0264
          Y=ACUMULADO(J)-1;                                                     ANA0265
          DO K=X TO Y;                                                          ANA0266
               S=ARGUMENTO(K);                                                  ANA0267
               N=ARGUMENTO(K+1);                                                ANA0268
               CALL CONSTRUIR_LISTA;                                            ANA0269
          END;                                                                  ANA0270
     END;                                                                       ANA0271
     S=NUMERO_DE_SIMBOLOS;                                                      ANA0272
     N=SIMBOLO_INICIAL;                                                         ANA0273
     CALL CONSTRUIR_LISTA;                                                      ANA0274
     S=SIMBOLO_INICIAL;                                                         ANA0275
     N=NUMERO_DE_SIMBOLOS;                                                      ANA0276
     CALL CONSTRUIR_LISTA;                                                      ANA0277
     FREE RAIZ;                                                                 ANA0278
     FREE ARGUMENTO;                                                            ANA0279
     FREE ACUMULADO;                                                            ANA0280
1/*************************************************************************/ANA0281
 /********************* CONSTRUIR LISTA (2.2.4.5) ************************/ANA0282
 /*************************************************************************/ANA0283
     CONSTRUIR_LISTA:                                                           ANA0284
     PROC;                                                                      ANA0285
0    IF CABECA_DE_LISTA(S)=0                                                    ANA0286
     THEN CABECA_DE_LISTA(S)=DISPONIVEL;                                        ANA0287
     ELSE                                                                       ANA0288
     DO;                                                                        ANA0289
          T=CABECA_DE_LISTA(S);                                                 ANA0290
          DO WHILE(T>0);                                                        ANA0291
               I=T;                                                             ANA0292
               T=PONTEIRO_IGUAL(T);                                             ANA0293
          END;                                                                  ANA0294
          PONTEIRO_IGUAL(I)=DISPONIVEL;                                         ANA0295
```

```
    END;                                                                ANA0296
    CONTRADOMINIO_IGUAL(DISPONIVEL)=N;                                  ANA0297
    PONTEIRO_IGUAL(DISPONIVEL)=0;                                       ANA0298
    DISPONIVEL=DISPONIVEL+1;                                            ANA0299
0   END;   /*** FIM DE CONSTRUIR_LISTA ***/                            ANA0300
1/*****************************************************************/   ANA0301
 /*************** CONSTRUIR GRAFO (2.2.4.6) ***********************/   ANA0302
 /*****************************************************************/   ANA0303
    CONSTRUIR_GRAFO:                                                    ANA0304
    PROC (CONTRADOMINIO);                                               ANA0305
    DCL CONTRADOMINIO(*)                    BIN FIXED;                  ANA0306
0   X=0;                                                                ANA0307
    DO S=1 TO NAOTERMINAIS;                                             ANA0308
        T=CABECA_DE_LISTA(S);                                           ANA0309
        DO WHILE (T>0);                                                 ANA0310
            X=X+1;                                                      ANA0311
            CONTRADOMINIO(X)=CONTRADOMINIO_IGUAL(T);                    ANA0312
            T=PONTEIRO_IGUAL(T);                                        ANA0313
        END;                                                            ANA0314
        CABECA_DE_LISTA(S)=X;                                           ANA0315
    END;                                                                ANA0316
0   END;   /*** FIM DE CONSTRUIR_GRAFO ***/                            ANA0317
0   END;   /*** FIM DE CONSTRUIR_IGUAL_PRIMEIRO_ULTIMO ***/            ANA0318
0   END;   /*** FIM DE TRATAR_PRODUCOES ***/                          ANA0319
1/*****************************************************************/   ANA0320
 /*************** CONSTRUIR MENOR E MAIOR (2.3) *******************/   ANA0321
 /*****************************************************************/   ANA0322
    CONSTRUIR_MENOR_MAIOR:                                              ANA0323
    PROC;                                                               ANA0324
    DCL DIREITOS(1)                         BIN FIXED CONTROLLED,       ANA0325
        ESQUERDOS(1)                        BIN FIXED CONTROLLED,       ANA0326
        NUMERO_DE_ESQUERDOS                 BIN FIXED,                  ANA0327
        NUMERO_DE_DIREITOS                  BIN FIXED,                  ANA0328
        MASCARA_INDICE_EXTREMOS(16)         BIN FIXED,                  ANA0329
        INDICE_EXTREMOS(256)                BIT(1) BASED(ADDR           ANA0330
                                            (MASCARA_INDICE_EXTREMOS)), ANA0331
        SIMBOLO                             BIN FIXED,                  ANA0332
        MENOR                               BIN FIXED INIT(1),          ANA0333
        IGUAL                               BIN FIXED INIT(2),          ANA0334
        MAIOR                               BIN FIXED INIT(3);          ANA0335
0   DCL REGISTRO                            CHAR(80),                   ANA0336
        1 RELACAO(13)                       BASED(ADDR(REGISTRO)),      ANA0337
          2 A                               BIN FIXED,                  ANA0338
          2 B                               BIN FIXED,                  ANA0339
          2 C                               BIN FIXED,                  ANA0340
        ARQREL                              FILE RECORD OUTPUT;         ANA0341
0        /*** ALOCAR AREAS DE TRABALHO (2.3.1) ***/                    ANA0342
    ALLOCATE ESQUERDOS(NUMERO_DE_SIMBOLOS);                             ANA0343
    ALLOCATE DIREITOS(NUMERO_DE_SIMBOLOS);                              ANA0344
    X=0;                                                                ANA0345
1   DO K=1 TO NUMERO_DE_SIMBOLOS;                                       ANA0346
        NUMERO_DE_ESQUERDOS, NUMERO_DE_DIREITOS = 0;                    ANA0347
        DO J=1 TO 16;                                                   ANA0348
            MASCARA_INDICE_EXTREMOS(J)=0;                               ANA0349
        END;                                                            ANA0350
0       /*** GERAR ESQUERDOS CONSTRUINDO PRIMEIRO+ (2.3.2) ***/        ANA0351
        J=IGUAL;                                                        ANA0352
        IF K=NUMERO_DE_SIMBOLOS                                         ANA0353
        THEN J=MENOR;                                                   ANA0354
        T=DOMINIO_IGUAL(K);                                             ANA0355
```

```
      DO WHILE(T>0);                                                    ANA0356
           SIMBOLO=CONTRADOMINIO_IGUAL(T);                              ANA0357
           IF SIMBOLO=NUMERO_DE_SIMBOLOS                                ANA0358
           THEN J=MAIOR;                                                ANA0359
           CALL GRAVAR_RELACOES(K,SIMBOLO,J);                           ANA0360
           IF SIMBOLO>NAOTERMINAIS                                      ANA0361
           THEN;                                                        ANA0362
           ELSE                                                         ANA0363
           CALL CONSTRUIR_FECHAMENTO_TRANSITIVO (DOMINIO_PRIMEIRO,      ANA0364
           CONTRADOMINIO_PRIMEIRO,ESQUERDOS,NUMERO_DE_ESQUERDOS);       ANA0365
           T=PONTEIRO_IGUAL(T);                                         ANA0366
      END;                                                              ANA0367
0     /*** CONSTRUIR MENOR = IGUAL X PRIMEIRO+ (2.3.3) ***/            ANA0368
      DO J=1 TO NUMERO_DE_ESQUERDOS;                                    ANA0369
           CALL GRAVAR_RELACOES(K,ESQUERDOS(J),MENOR);                  ANA0370
      END;                                                              ANA0371
      T=DOMINIO_IGUAL(K);                                               ANA0372
      DO WHILE (T>0);                                                   ANA0373
           S=CONTRADOMINIO_IGUAL(T);                                    ANA0374
           IF INDICE_EXTREMOS(S)='0'B                                   ANA0375
           THEN                                                         ANA0376
           DO;                                                          ANA0377
                NUMERO_DE_ESQUERDOS=NUMERO_DE_ESQUERDOS+1;              ANA0378
                ESQUERDOS(NUMERO_DE_ESQUERDOS)=S;                       ANA0379
                INDICE_EXTREMOS(S)='1'B;                                ANA0380
           END;                                                         ANA0381
           T=PONTEIRO_IGUAL(T);                                         ANA0382
      END;                                                              ANA0383
0     /*** GERAR DIREITOS CONSTRUINDO ULTIMO+ (2.3.4) ***/             ANA0384
      DO J=1 TO 16;                                                     ANA0385
           MASCARA_INDICE_EXTREMOS(J)=0;                                ANA0386
      END;                                                              ANA0387
      SIMBOLO=K;                                                        ANA0388
      IF SIMBOLO>NAOTERMINAIS OR DOMINIO_IGUAL(SIMBOLO)=0               ANA0389
      THEN;                                                             ANA0390
      ELSE CALL CONSTRUIR_FECHAMENTO_TRANSITIVO(DOMINIO_ULTIMO,         ANA0391
      CONTRADOMINIO_ULTIMO,DIREITOS,NUMERO_DE_DIREITOS);                ANA0392
0     /*** CONSTRUIR MAIOR = I(ULTIMO)+ X I(PRIMEIRO)+                 ANA0393
                                                  (2.3.5) ***/          ANA0394
      DO J=1 TO NUMERO_DE_DIREITOS;                                     ANA0395
           DO N=1 TO NUMERO_DE_ESQUERDOS;                               ANA0396
                IF ESQUERDOS(N)>NAOTERMINAIS                            ANA0397
                THEN                                                    ANA0398
                CALL GRAVAR_RELACOES(DIREITOS(J),ESQUERDOS(N),MAIOR);   ANA0399
           END;                                                         ANA0400
      END;                                                              ANA0401
 END;                                                                   ANA0402
 K=0;                                                                   ANA0403
 CALL GRAVAR_RELACOES(K,S,J);                                           ANA0404
 X=13;                                                                  ANA0405
 CALL GRAVAR_RELACOES(K,S,J);                                           ANA0406
1/*****************************************************************/ANA0407
 /************** CONSTRUIR FECHAMENTO TRANSITIVO (2.3.6) **************/ANA0408
 /*****************************************************************/ANA0409
 CONSTRUIR_FECHAMENTO_TRANSITIVO:                                       ANA0410
 PROC (DOMINIO, CONTRADOMINIO, EXTREMOS, NUMERO_DE_EXTREMOS);           ANA0411
 DCL DOMINIO(*)                    BIN FIXED ,                          ANA0412
     CONTRADOMINIO(*)              BIN FIXED ,                          ANA0413
     EXTREMOS(*)                   BIN FIXED ,                          ANA0414
     NUMERO_DE_EXTREMOS            BIN FIXED,                           ANA0415
```

```
      (S,P,N,J,T)                          BIN FIXED;                    ANA0416
0     S=NUMERO_DE_EXTREMOS;                                              ANA0417
      P=NUMERO_DE_EXTREMOS+1;                                            ANA0418
      J=DOMINIO(SIMBOLO-1)+1;                                            ANA0419
      DO N=J TO DOMINIO(SIMBOLO);                                        ANA0420
           IF INDICE_EXTREMOS(CONTRADOMINIO(N))='0'B                     ANA0421
           THEN                                                          ANA0422
           DO;                                                           ANA0423
                S=S+1;                                                   ANA0424
                EXTREMOS(S)=CONTRADOMINIO(N);                            ANA0425
                INDICE_EXTREMOS(CONTRADOMINIO(N))='1'B;                  ANA0426
           END;                                                          ANA0427
      END;                                                               ANA0428
      DO WHILE (P<=S);                                                   ANA0429
           IF EXTREMOS(P)>NAOTERMINAIS                                   ANA0430
           THEN;                                                         ANA0431
           ELSE                                                          ANA0432
           DO;                                                           ANA0433
                J=DOMINIO(EXTREMOS(P)-1)+1;                              ANA0434
                T=DOMINIO(EXTREMOS(P));                                  ANA0435
                DO N=J TO T;                                             ANA0436
                     IF INDICE_EXTREMOS(CONTRADOMINIO(N))='0'B           ANA0437
                     THEN                                                ANA0438
                     DO;                                                 ANA0439
                          S=S+1;                                         ANA0440
                          EXTREMOS(S)=CONTRADOMINIO(N);                  ANA0441
                          INDICE_EXTREMOS(CONTRADOMINIO(N))='1'B;        ANA0442
                     END;                                                ANA0443
                END;                                                     ANA0444
           END;                                                          ANA0445
           P=P+1;                                                        ANA0446
      END;                                                               ANA0447
      NUMERO_DE_EXTREMOS=S;                                              ANA0448
0     END;    /*** FIM DE CONSTRUIR_FECHAMENTO_TRANSITIVO ***/           ANA0449
1/*****************************************************************/     ANA0450
 /********************* GRAVAR RELACOES (2.3.7) *********************/     ANA0451
 /*****************************************************************/     ANA0452
      GRAVAR_RELACOES:                                                   ANA0453
      PROC (PRIMEIRO,SEGUNDO,PRECEDENCIA);                               ANA0454
0     DCL PRIMEIRO                         BIN FIXED,                    ANA0455
          SEGUNDO                          BIN FIXED,                    ANA0456
          PRECEDENCIA                      BIN FIXED;                    ANA0457
0     X=X+1;                                                             ANA0458
      IF X>13                                                            ANA0459
      THEN                                                               ANA0460
      DO;                                                                ANA0461
           WRITE FILE (ARQREL) FROM (REGISTRO);                          ANA0462
           X=1;                                                          ANA0463
      END;                                                               ANA0464
0     RELACAO(X).A=PRIMEIRO;                                             ANA0465
      RELACAO(X).B=SEGUNDO;                                              ANA0466
      RELACAO(X).C=PRECEDENCIA;                                          ANA0467
0     END;    /*** FIM DE GRAVAR_RELACOES ***/                          ANA0468
0     END;    /** FIM DE CONSTRUIR_MENOR_MAIOR ***/                      ANA0469
1/*****************************************************************/     ANA0470
 /******************** GRAVAR CONTROLES (2.4) ********************/     ANA0471
 /*****************************************************************/     ANA0472
      GRAVAR_CONTROLES:                                                  ANA0473
      PROC(ERRO);                                                        ANA0474
      DCL ERRO                             CHAR(1),                      ANA0475
```

```
        REGISTRO                        CHAR(80),                       ANA0476
        CARACTER(80)                    CHAR(1) DEF REGISTRO,           ANA0477
        CONTROLES(40)                   BIN FIXED BASED                 ANA0478
                                        (ADDR(REGISTRO)),               ANA0479
        ARQCTL1                         FILE RECORD OUTPUT;             ANA0480
0   IF ERRO=BRANCO                                                      ANA0481
    THEN CONTROLES(1)=0;                                                ANA0482
    ELSE CARACTER(1)=ERRO;                                              ANA0483
    CONTROLES(2)=NAOTERMINAIS;                                          ANA0484
    CONTROLES(3)=NUMERO_DE_SIMBOLOS;                                    ANA0485
    CONTROLES(4)=NUMERO_DE_PRODUCOES;                                   ANA0486
    CONTROLES(5)=TAMANHO_DOS_ARGUMENTOS;                                ANA0487
    CONTROLES(6)=SIMBOLO_INICIAL;                                       ANA0488
    WRITE FILE (ARQCTL1) FROM (REGISTRO);                               ANA0489
0   END;    /*** FIM DE GRAVAR_ERRO ***/                              ANA0490
-   END;    /*** FIM DO PROGRAMA ANAG102 ***/                         ANA0491
```

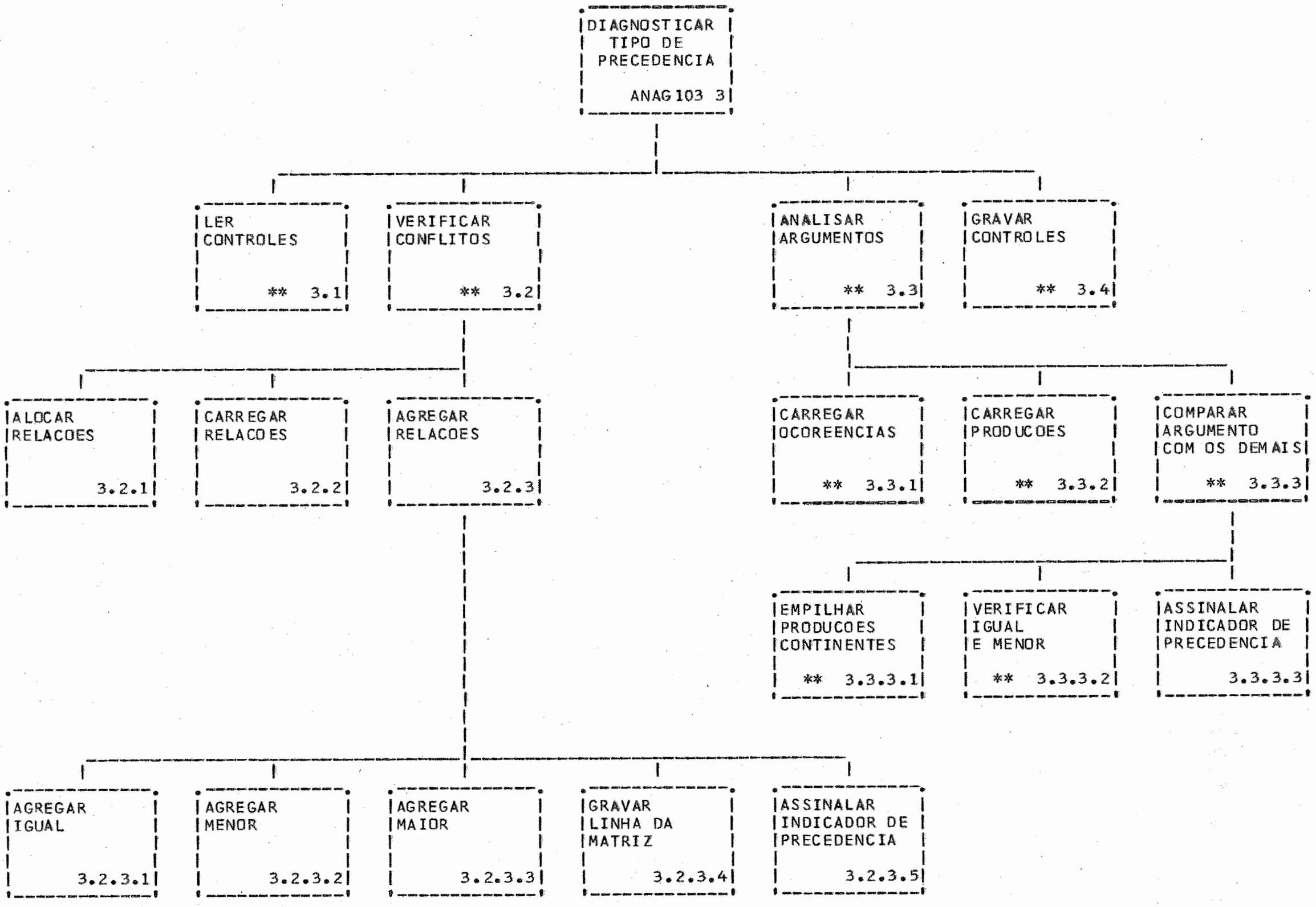

DIAGNOSTICAR TIPO DE PRECEDENCIA
ANAG103 3
LER CONTROLES
** 3.1
VERIFICAR CONFLITOS
** 3.2
ANALISAR ARGUMENTOS
** 3.3
GRAVAR CONTROLES
** 3.4
ALOCAR RELACOES
3.2.1
CARREGAR RELACOES
3.2.2
AGREGAR RELACOES
3.2.3
CARREGAR OCOREENCIAS
** 3.3.1
CARREGAR PRODUCOES
** 3.3.2
COMPARAR ARGUMENTO COM OS DEMAIS
** 3.3.3
EMPILHAR PRODUCOES CONTINENTES
** 3.3.3.1
VERIFICAR IGUAL E MENOR
** 3.3.3.2
ASSINALAR INDICADOR DE PRECEDENCIA
3.3.3.3
AGREGAR IGUAL
3.2.3.1
AGREGAR MENOR
3.2.3.2
AGREGAR MAIOR
3.2.3.3
GRAVAR LINHA DA MATRIZ
3.2.3.4
ASSINALAR INDICADOR DE PRECEDENCIA
3.2.3.5

FIGURA IV.11

ANAG103 - DIAGNOSTICAR TIPO DE PRECEDENCIA (3)

ENTRADA

ARQUIVO 1
DE CONTROLE
(ARQCTL1)

RELACOES DE
PRECEDENCIA
(ARQREL)

NAO TERMINAIS
NUMERO DE SIMBOLOS
NUMERO DE PRODUCOES

PROCESSO

01 LER CONTROLES.

02 VERIFICAR CONFLITOS.

AS TRES RELACOES SAO LIDAS DO DISCO E FICAM REPRESENTADAS SOB A FORMA DE LISTAS .AS LISTAS DE CADA RELACAO PARA CADA SIMBOLO SAO PERCORRIDAS E UMA LINHA DA MATRIZ DE PRECEDENCIA E' FORMATADA. DESTA FORMA QDO. AS LISTAS DO SIMBOLO I SAO PERCORRIDAS ENTAO A LINHA I ESTARA' SENDO FORMATADA. OS CONFLITOS DE PRECEDENCIA SAO TESTADOS E CODIFICADOS. A CODIFICACAO DOS VALORES DE CADA POSICAO DA LINHA SAO OS SEGUINTES:
1=MENOR, 2=IGUAL, 3=MAIOR,
4=MENOR/IGUAL, 5=MAIOR/IGUAL,
6=MENOR/MAIOR, 7=IGUAL/MENOR/MAIOR. E' GERADO UM INDICADOR DE PRECEDENCIA , QUE TERA' VALOR 0 SE NAO HOUVER CONFLITOS, QUE TERA' VALOR 1 SE HOUVER CONFLITOS MENOR/IGUAL E VALOR 2 SE HOUVER CONFLITOS ENVOLVENDO MAIOR. CADA LINHA APOS SER FORMATADA E' GRAVADA EM DISCO EM UM ARQUIVO DE ACESSO DIRETO CUJA CHAVE DE ACESSO E' O NUMERO DA LINHA, GERANDO ASSIM A MATRIZ DE PRECEDENCIA.

SAIDA

CONTROLES
NAO TERMINAIS
NUMERO DE SIMBOLOS
NUMERO DE PRODUCOES

MATRIZ DE
PRECEDENCIA
(ARQMAT)

INDICADOR PRECEDENCIA

RELACAO IGUAL
RELACAO MENOR

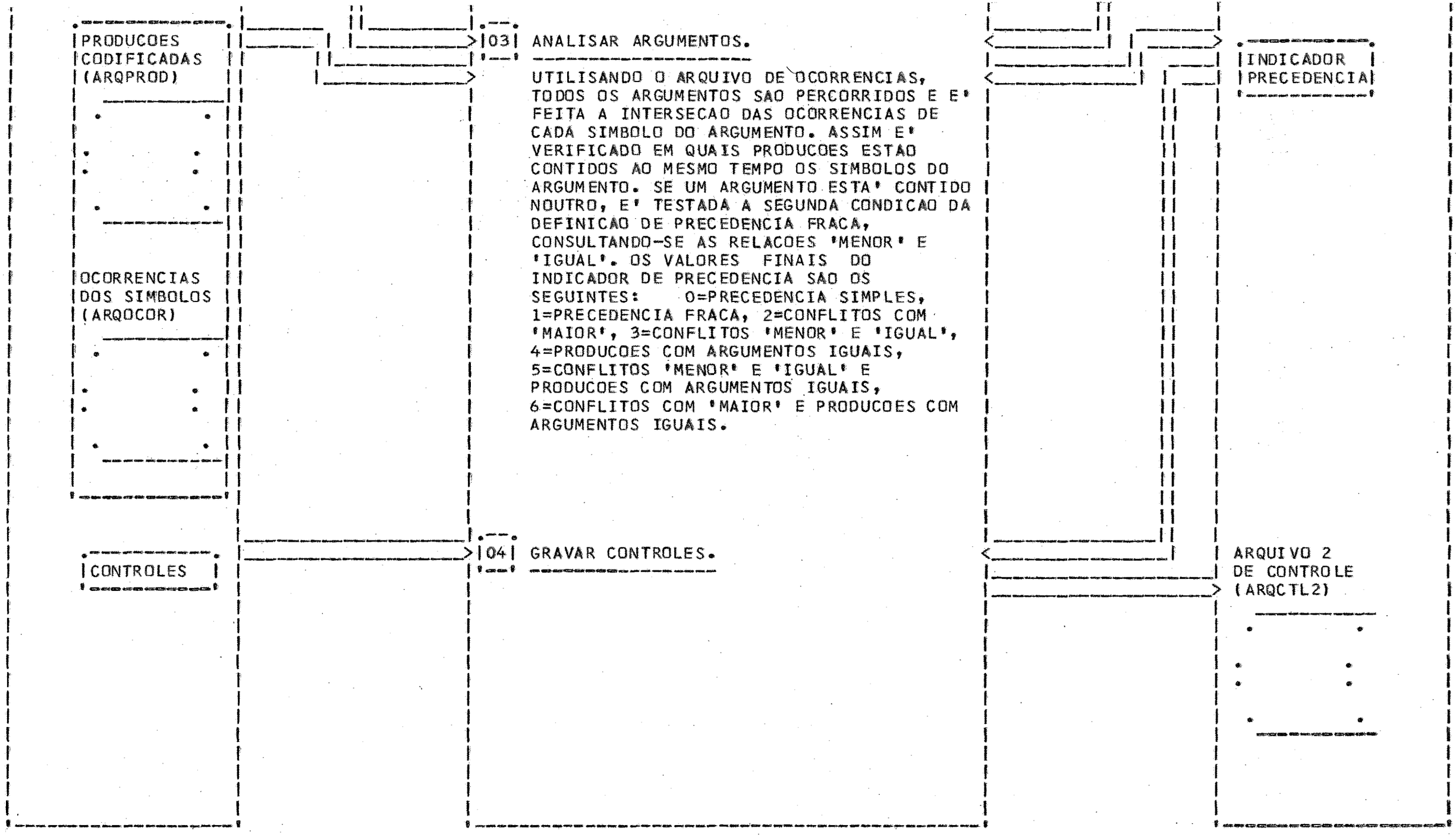

PRODUCOES CODIFICADAS (ARQPROD)
OCORRENCIAS DOS SIMBOLOS (ARQOCOR)
03 ANALISAR ARGUMENTOS.
UTILISANDO O ARQUIVO DE OCORRENCIAS, TODOS OS ARGUMENTOS SAO PERCORRIDOS E E' FEITA A INTERSECAO DAS OCORRENCIAS DE CADA SIMBOLO DO ARGUMENTO. ASSIM E' VERIFICADO EM QUAIS PRODUCOES ESTAO CONTIDOS AO MESMO TEMPO OS SIMBOLOS DO ARGUMENTO. SE UM ARGUMENTO ESTA' CONTIDO NOUTRO, E' TESTADA A SEGUNDA CONDICAO DA DEFINICAO DE PRECEDENCIA FRACA, CONSULTANDO-SE AS RELACOES 'MENOR' E 'IGUAL'. OS VALORES FINAIS DO INDICADOR DE PRECEDENCIA SAO OS SEGUINTES: 0=PRECEDENCIA SIMPLES, 1=PRECEDENCIA FRACA, 2=CONFLITOS COM 'MAIOR', 3=CONFLITOS 'MENOR' E 'IGUAL', 4=PRODUCOES COM ARGUMENTOS IGUAIS, 5=CONFLITOS 'MENOR' E 'IGUAL' E PRODUCOES COM ARGUMENTOS IGUAIS, 6=CONFLITOS COM 'MAIOR' E PRODUCOES COM ARGUMENTOS IGUAIS.
INDICADOR PRECEDENCIA
04 GRAVAR CONTROLES.
CONTROLES
ARQUIVO 2 DE CONTROLE (ARQCTL2)

FIGURA IV.12

```
         ****************************************
         *  LISTAGEM FONTE DO PROGRAMA ANAG103  *
         ****************************************


 /* DIAGNOSTICAR TIPO DE PRECEDENCIA (3)  (VERSAO 07/12/78)          */ANA0001
-    ANAG103:                                                          ANA0002
     PROC OPTIONS(MAIN);                                              ANA0003
0    DCL NAOTERMINAIS                        BIN FIXED,               ANA0004
         NUMERO_DE_SIMBOLOS                  BIN FIXED,               ANA0005
         NUMERO_DE_PRODUCOES                 BIN FIXED,               ANA0006
         TAMANHO_DOS_ARGUMENTOS              BIN FIXED,               ANA0007
         SIMBOLO_INICIAL                     BIN FIXED,               ANA0008
         DOMINIO_IGUAL(0:1)                  BIN FIXED CONTROLLED,    ANA0009
         CONTRADOMINIO_IGUAL(1)              BIN FIXED CONTROLLED,    ANA0010
         DOMINIO_MENOR(0:1)                  BIN FIXED CONTROLLED,    ANA0011
         CONTRADOMINIO_MENOR(1)              BIN FIXED CONTROLLED,    ANA0012
         INDICADOR_DE_PRECEDENCIA            BIN FIXED INIT(0),       ANA0013
         ERRO                                CHAR(1) INIT(' ');       ANA0014
0    /** VARIAVEIS GLOBAIS DE TRABALHO **/                            ANA0015
     DCL BRANCO                              CHAR(1) INIT(' '),       ANA0016
         (I,J,K,N,X,Y,S,T)                   BIN FIXED;               ANA0017
-    CALL LER_CONTROLES;                                               ANA0018
     IF ERRO = BRANCO                                                 ANA0019
     THEN                                                             ANA0020
     DO;                                                              ANA0021
          CALL VERIFICAR_CONFLITOS;                                   ANA0022
          CALL ANALISAR_ARGUMENTOS;                                   ANA0023
     END;                                                             ANA0024
     ELSE;                                                            ANA0025
          CALL GRAVAR_CONTROLES;                                      ANA0026
1/*****************************************************************/  ANA0027
 /******************** LER CONTROLES (3.1) ************************/  ANA0028
 /*****************************************************************/  ANA0029
     LER_CONTROLES:                                                   ANA0030
     PROC;                                                            ANA0031
0    DCL P                                   POINTER,                 ANA0032
         CONTROLES(40)                       BIN FIXED BASED(P),      ANA0033
         CARATER(80)                         CHAR(1) BASED(P),        ANA0034
         ARQCTL1                             FILE RECORD INPUT;       ANA0035
0    READ FILE (ARQCTL1) SET (P);                                     ANA0036
     IF CONTROLES(1)=0                                                ANA0037
     THEN;                                                            ANA0038
     ELSE ERRO=CARATER(1);                                            ANA0039
0    NAOTERMINAIS=CONTROLES(2);                                       ANA0040
     NUMERO_DE_SIMBOLOS=CONTROLES(3);                                 ANA0041
     NUMERO_DE_PRODUCOES=CONTROLES(4);                                ANA0042
     TAMANHO_DOS_ARGUMENTOS=CONTROLES(5);                             ANA0043
     SIMBOLO_INICIAL=CONTROLES(6);                                    ANA0044
0    CLOSE FILE (ARQCTL1);                                            ANA0045
0    END;   /*** FIM DE LER_CONTROLES ***/                            ANA0046
1/*****************************************************************/  ANA0047
 /***************** VERIFICAR CONFLITOS (3.2) **********************/  ANA0048
 /*****************************************************************/  ANA0049
     VERIFICAR_CONFLITOS:                                             ANA0050
     PROC;                                                            ANA0051
 DCL ARQREL                              FILE RECORD INPUT,           ANA0052
      P                                  POINTER,                     ANA0053
      1 RELACAO(13)                      BASED(P),                    ANA0054
        2 A                              BIN FIXED,                   ANA0055
```

```
      2 B                                  BIN FIXED,                          ANA0056
      2 C                                  BIN FIXED;                          ANA0057
ODCL  PRIMEIRO                             BIN FIXED,                          ANA0058
      SEGUNDO                              BIN FIXED,                          ANA0059
      PRECEDENCIA                          BIN FIXED,                          ANA0060
      DISPONIVEL_IGUAL                     BIN FIXED INIT(1),                  ANA0061
      DISPONIVEL_MENOR                     BIN FIXED INIT(1),                  ANA0062
      DISPONIVEL_MAIOR                     BIN FIXED INIT(1),                  ANA0063
      SIM                                  CHAR(1) INIT('1'),                  ANA0064
      NAO                                  CHAR(1) INIT('0'),                  ANA0065
      REPETICAO                            CHAR(1) INIT('0');                  ANA0066
ODCL  DOMINIO_MAIOR(0:1)                   BIN FIXED CONTROLLED,               ANA0067
      CONTRADOMINIO_MAIOR(1)               BIN FIXED CONTROLLED,               ANA0068
      PONTEIRO_MAIOR(1)                    BIN FIXED CONTROLLED,               ANA0069
      X                                    DEC FIXED(3),                       ANA0070
      CHAVE                                PIC '(8)9',                         ANA0071
      MENOR                                BIN FIXED INIT(1),                  ANA0072
      IGUAL                                BIN FIXED INIT(2),                  ANA0073
      MAIOR                                BIN FIXED INIT(3);                  ANA0074
ODCL  TABELA_DE_CONFLITOS_COM_MAIOR(0:4)   BIN FIXED INIT(3,6,5,3,7),          ANA0075
      LINHA(256)                           BIN FIXED INIT((255)(0)),           ANA0076
      LINHA_CARAC(512)                     CHAR(1) BASED(ADDR(LINHA)),         ANA0077
      REGISTRO                             CHAR(256),                          ANA0078
      LINHA_DA_MATRIZ(256)                 CHAR(1) BASED(ADDR(REGISTRO)),      ANA0079
      ARQMAT                               FILE RECORD DIRECT OUTPUT           ANA0080
                                           KEYED ENV(REGIONAL(1));             ANA0081
1          /*** ALOCAR RELACOES (3.2.1) ***/                                   ANA0082
     J=4*(TAMANHO_DOS_ARGUMENTOS + NUMERO_DE_PRODUCOES);                       ANA0083
     K=TAMANHO_DOS_ARGUMENTOS - NUMERO_DE_PRODUCOES;                           ANA0084
     ALLOCATE DOMINIO_MENOR(0:NUMERO_DE_SIMBOLOS) INIT(0);                     ANA0085
     ALLOCATE CONTRADOMINIO_MENOR(J);                                          ANA0086
     ALLOCATE DOMINIO_IGUAL(0:NUMERO_DE_SIMBOLOS) INIT(0);                     ANA0087
     ALLOCATE CONTRADOMINIO_IGUAL(K);                                          ANA0088
     ALLOCATE DOMINIO_MAIOR(NUMERO_DE_SIMBOLOS);                               ANA0089
     ALLOCATE CONTRADOMINIO_MAIOR(J);                                          ANA0090
     ALLOCATE PONTEIRO_MAIOR(J);                                               ANA0091
     DO J=1 TO NUMERO_DE_SIMBOLOS;                                             ANA0092
          DOMINIO_MENOR(J),DOMINIO_IGUAL(J),DOMINIO_MAIOR(J)=0;                ANA0093
     END;                                                                      ANA0094
     X=13;                                                                     ANA0095
     K=0;                                                                      ANA0096
1         /*** CARREGAR RELACOES (3.2.2) ***/                                 ANA0097
     CALL LER_RELACOES;                                                        ANA0098
     DO WHILE(PRIMEIRO>0);                                                     ANA0099
0         /*** CARREGAR MENOR (3.2.2.1) ***/                                  ANA0100
          IF PRECEDENCIA=IGUAL                                                 ANA0101
          THEN                                                                 ANA0102
          DO;                                                                  ANA0103
               DOMINIO_IGUAL(PRIMEIRO)=DISPONIVEL_IGUAL;                       ANA0104
               CONTRADOMINIO_IGUAL(DISPONIVEL_IGUAL)=SEGUNDO;                  ANA0105
               DISPONIVEL_IGUAL=DISPONIVEL_IGUAL+1;                            ANA0106
          END;                                                                 ANA0107
0         /*** CARREGAR IGUAL (3.2.2.2) ***/                                  ANA0108
          IF PRECEDENCIA=MENOR                                                 ANA0109
          THEN                                                                 ANA0110
          DO;                                                                  ANA0111
               DOMINIO_MENOR(PRIMEIRO)=DISPONIVEL_MENOR;                       ANA0112
               CONTRADOMINIO_MENOR(DISPONIVEL_MENOR)=SEGUNDO;                  ANA0113
               DISPONIVEL_MENOR=DISPONIVEL_MENOR+1;                            ANA0114
          END;                                                                 ANA0115
```

```
0       /*** CARREGAR MAIOR (3.2.2.3) ***/                                  ANA0116
        IF PRECEDENCIA=MAIOR                                               ANA0117
        THEN                                                               ANA0118
        DO;                                                                ANA0119
             IF DOMINIO_MAIOR(PRIMEIRO)=0                                  ANA0120
             THEN DOMINIO_MAIOR(PRIMEIRO)=DISPONIVEL_MAIOR;                ANA0121
             ELSE                                                          ANA0122
             DO;                                                           ANA0123
                  T=DOMINIO_MAIOR(PRIMEIRO);                               ANA0124
                  DO WHILE(T>0);                                           ANA0125
                       IF CONTRADOMINIO_MAIOR(T)=SEGUNDO                   ANA0126
                       THEN REPETICAO=SIM;                                 ANA0127
                       I=T;                                                ANA0128
                       T=PONTEIRO_MAIOR(T);                                ANA0129
                  END;                                                     ANA0130
                  PONTEIRO_MAIOR(I)=DISPONIVEL_MAIOR;                      ANA0131
             END;                                                          ANA0132
             IF REPETICAO=NAO                                              ANA0133
             THEN                                                          ANA0134
             DO;                                                           ANA0135
                  CONTRADOMINIO_MAIOR(DISPONIVEL_MAIOR)=SEGUNDO;           ANA0136
                  PONTEIRO_MAIOR(DISPONIVEL_MAIOR)=0;                      ANA0137
                  DISPONIVEL_MAIOR=DISPONIVEL_MAIOR+1;                     ANA0138
             END;                                                          ANA0139
             ELSE                                                          ANA0140
             DO;                                                           ANA0141
                  PONTEIRO_MAIOR(I)=0;                                     ANA0142
                  REPETICAO=NAO;                                           ANA0143
             END;                                                          ANA0144
        END;                                                               ANA0145
             CALL LER_RELACOES;                                            ANA0146
        END;                                                               ANA0147
        DO J=1 TO NUMERO_DE_SIMBOLOS;                                      ANA0148
             IF DOMINIO_IGUAL(J)=0                                         ANA0149
             THEN DOMINIO_IGUAL(J)=DOMINIO_IGUAL(J-1);                     ANA0150
             IF DOMINIO_MENOR(J)=0                                         ANA0151
             THEN DOMINIO_MENOR(J)=DOMINIO_MENOR(J-1);                     ANA0152
        END;                                                               ANA0153
        CLOSE FILE(ARQREL);                                                ANA0154
I            /*** AGREGAR RELACOES (3.2.3) ***/                           ANA0155
0       N=0;                                                               ANA0156
        DO X=1 TO NUMERO_DE_SIMBOLOS;                                      ANA0157
0            /*** AGREGAR IGUAL (3.2.3.1) ***/                            ANA0158
             S=DOMINIO_IGUAL(X-1)+1;                                       ANA0159
             J=DOMINIO_IGUAL(X);                                           ANA0160
             DO K=S TO J;                                                  ANA0161
                  LINHA(CONTRADOMINIO_IGUAL(K))=2;                         ANA0162
             END;                                                          ANA0163
0            /*** AGREGAR MENOR (3.2.3.2) ***/                            ANA0164
             S=DOMINIO_MENOR(X-1)+1;                                       ANA0165
             J=DOMINIO_MENOR(X);                                           ANA0166
             DO K=S TO J;                                                  ANA0167
                  I=CONTRADOMINIO_MENOR(K);                                ANA0168
                  IF LINHA(I)=0                                            ANA0169
                  THEN LINHA(I)=1;                                         ANA0170
                  ELSE IF LINHA(I)=2                                       ANA0171
                       THEN                                                ANA0172
                       DO;                                                 ANA0173
                            LINHA(I)=4;                                    ANA0174
                            INDICADOR_DE_PRECEDENCIA=1;                    ANA0175
```

```
                    END;                                                    ANA0176
                    ELSE;                                                   ANA0177
          END;                                                              ANA0178
0         /*** AGREGAR MAIOR (3.2.3.3) ***/                                ANA0179
          T=DOMINIO_MAIOR(X);                                               ANA0180
          DO WHILE(T>0);                                                    ANA0181
               S=CONTRADOMINIO_MAIOR(T);                                    ANA0182
               LINHA(S)=TABELA_DE_CONFLITOS_COM_MAIOR(LINHA(S));            ANA0183
               IF LINHA(S)>3                                                ANA0184
               THEN INDICADOR_DE_PRECEDENCIA=2;                             ANA0185
               T=PONTEIRO_MAIOR(T);                                         ANA0186
          END;                                                              ANA0187
0         /*** GRAVAR LINHA DA MATRIZ (3.2.3.4) ***/                       ANA0188
          DO N=1 TO NUMERO_DE_SIMBOLOS;                                     ANA0189
               LINHA_DA_MATRIZ(N)=LINHA_CARAC(2*N);                         ANA0190
          END;                                                              ANA0191
          CHAVE = X - 1;                                                    ANA0192
          WRITE FILE (ARQMAT) FROM (REGISTRO) KEYFROM (CHAVE);              ANA0193
          DO K=1 TO NUMERO_DE_SIMBOLOS;                                     ANA0194
               LINHA(K)=0;                                                  ANA0195
          END;                                                              ANA0196
     END;                                                                   ANA0197
     FREE DOMINIO_MAIOR;                                                    ANA0198
     FREE CONTRADOMINIO_MAIOR;                                              ANA0199
     FREE PONTEIRO_MAIOR;                                                   ANA0200
1/*****************************************************************/       ANA0201
 /******************** LER RELACOES (3.2.2.4) **********************/       ANA0202
 /*****************************************************************/       ANA0203
 LER_RELACOES:                                                              ANA0204
 PROC;                                                                      ANA0205
     X=X+1;                                                                 ANA0206
     IF X>13                                                                ANA0207
     THEN                                                                   ANA0208
     DO;                                                                    ANA0209
          READ FILE (ARQREL) SET (P);                                       ANA0210
          X=1;                                                              ANA0211
     END;                                                                   ANA0212
     PRIMEIRO=A(X);                                                         ANA0213
     SEGUNDO=B(X);                                                          ANA0214
     PRECEDENCIA=C(X);                                                      ANA0215
0    END;    /*** FIM DE LER_RELACOES ***/                                 ANA0216
0    END;    /*** FIM DE VERIFICAR_CONFLITOS ***/                          ANA0217
1/*****************************************************************/       ANA0218
 /******************** ANALISAR ARGUMENTOS (3.3) ******************/       ANA0219
 /*****************************************************************/       ANA0220
     ANALISAR_ARGUMENTOS:                                                   ANA0221
     PROC;                                                                  ANA0222
     DCL RAIZ(1)                          BIN FIXED CONTROLLED,             ANA0223
         ARGUMENTO(1)                     BIN FIXED CONTROLLED,             ANA0224
         ACUMULADO(0:1)                   BIN FIXED CONTROLLED,             ANA0225
         DOMINIO_OCORRE(0:1)              BIN FIXED CONTROLLED,             ANA0226
         CONTRADOMINIO_OCORRE(1)          BIN FIXED CONTROLLED,             ANA0227
         ANTIGO_NUMERO_DE_SIMBOLOS        BIN FIXED,                        ANA0228
         P                                POINTER,                          ANA0229
         VETOR(40)                        BIN FIXED BASED(P);               ANA0230
1/*****************************************************************/       ANA0231
 /********** CARREGAR OCORRENCIAS (3.3.1) **********************/          ANA0232
 /*****************************************************************/       ANA0233
     BEGIN;                                                                 ANA0234
     DCL ARQOCOR                          FILE RECORD INPUT;                ANA0235
```

```
0    J=NUMERO_DE_PRODUCOES+TAMANHO_DOS_ARGUMENTOS;                    ANA0236
     ALLOCATE DOMINIO_OCORRE(0:NUMERO_DE_SIMBOLOS) INIT(0);           ANA0237
     ALLOCATE CONTRADOMINIO_OCORRE(J);                                ANA0238
0    X=1;                                                             ANA0239
     K=0;                                                             ANA0240
     ANTIGO_NUMERO_DE_SIMBOLOS=NUMERO_DE_SIMBOLOS-1;                  ANA0241
     READ FILE (ARQOCOR) SET (P);                                     ANA0242
0    DO N=1 TO ANTIGO_NUMERO_DE_SIMBOLOS;                             ANA0243
          DO WHILE(VETOR(X)>0);                                       ANA0244
               K=K+1;                                                 ANA0245
               CONTRADOMINIO_OCORRE(K)=VETOR(X);                      ANA0246
               X=X+1;                                                 ANA0247
               IF X>40                                                ANA0248
               THEN                                                   ANA0249
               DO;                                                    ANA0250
                    READ FILE (ARQOCOR) SET (P);                      ANA0251
                    X=1;                                              ANA0252
               END;                                                   ANA0253
          END;                                                        ANA0254
          X=X+1;                                                      ANA0255
          IF X>40                                                     ANA0256
          THEN                                                        ANA0257
          DO;                                                         ANA0258
               READ FILE (ARQOCOR) SET (P);                           ANA0259
               X=1;                                                   ANA0260
          END;                                                        ANA0261
          DOMINIO_OCORRE(N)=K;                                        ANA0262
     END;                                                             ANA0263
     CLOSE FILE(ARQOCOR);                                             ANA0264
0    END;   /*** FIM DE CARREGAR OCORRENCIAS ***/                     ANA0265
1/*****************************************************************/  ANA0266
 /********** CARREGAR PRODUCOES (3.3.2) ***********************/      ANA0267
 /*****************************************************************/  ANA0268
     BEGIN;                                                           ANA0269
     DCL ARQPROD                                  FILE RECORD INPUT;  ANA0270
0    ALLOCATE RAIZ(NUMERO_DE_PRODUCOES);                              ANA0271
     ALLOCATE ARGUMENTO(TAMANHO_DOS_ARGUMENTOS);                      ANA0272
     ALLOCATE ACUMULADO(0:NUMERO_DE_PRODUCOES) INIT(0);               ANA0273
0    X=40;                                                            ANA0274
     Y,S,T=0;                                                         ANA0275
     DO K=1 TO 3;                                                     ANA0276
          CALL LER_ARQUIVO_CODIGOS;                                   ANA0277
     END;                                                             ANA0278
0    DO WHILE (VETOR(X)>0);                                           ANA0279
          Y=Y+1;                                                      ANA0280
          RAIZ(Y)=VETOR(X);                                           ANA0281
          CALL LER_ARQUIVO_CODIGOS;                                   ANA0282
          DO WHILE (VETOR(X)>0);                                      ANA0283
               S=S+1;                                                 ANA0284
               ARGUMENTO(S)=VETOR(X);                                 ANA0285
               CALL LER_ARQUIVO_CODIGOS;                              ANA0286
          END;                                                        ANA0287
          CALL LER_ARQUIVO_CODIGOS;                                   ANA0288
          T=T+1;                                                      ANA0289
          ACUMULADO(T)=VETOR(X);                                      ANA0290
          CALL LER_ARQUIVO_CODIGOS;                                   ANA0291
     END;                                                             ANA0292
     CLOSE FILE (ARQPROD);                                            ANA0293
1/*****************************************************************/  ANA0294
 /*************** LER ARQUIVO DE CODIGOS (3.3.2.1) ***************/   ANA0295
```

```
 /*****************************************************************/     ANA0296
    LER_ARQUIVO_CODIGOS:                                               ANA0297
    PROC;                                                              ANA0298
0   X=X+1;                                                             ANA0299
    IF X>40                                                            ANA0300
    THEN                                                               ANA0301
    DO;                                                                ANA0302
         READ FILE (ARQPROD) SET (P);                                  ANA0303
         X=1;                                                          ANA0304
    END;                                                               ANA0305
0   END;    /*** FIM DE LER_ARQUIVO_CODIGOS ***/                       ANA0306
0   END;    /*** FIM DE CARREGAR PRODUCOES ***/                        ANA0307
1/*****************************************************************/     ANA0308
 /********** COMPARAR UM ARGUMENTO COM OS DEMAIS (3.3.3) **********/     ANA0309
 /*****************************************************************/     ANA0310
    BEGIN;                                                             ANA0311
0   DCL SAIDA(1)                          BIT(1) CONTROLLED,           ANA0312
        ENTRADA(1)                        BIT(1) CONTROLLED,           ANA0313
        PILHA(1)                          BIN FIXED CONTROLLED,        ANA0314
        EMPILHADAS                        BIN FIXED,                   ANA0315
        EMPILHADAS_IGUAL                  BIN FIXED,                   ANA0316
        INICIO_ARGUMENTO                  BIN FIXED,                   ANA0317
        FIM_ARGUMENTO                     BIN FIXED,                   ANA0318
        PRODUCAO                          BIN FIXED,                   ANA0319
        TAMANHO                           BIN FIXED,                   ANA0320
        CONTIDO                           CHAR(1),                     ANA0321
        SIM                               CHAR(1) INIT('1'),           ANA0322
        NAO                               CHAR(1) INIT('0');           ANA0323
0   ALLOCATE ENTRADA(NUMERO_DE_PRODUCOES);                             ANA0324
    ALLOCATE SAIDA(NUMERO_DE_PRODUCOES);                               ANA0325
    ALLOCATE PILHA(NUMERO_DE_PRODUCOES);                               ANA0326
0   DO PRODUCAO=1 TO NUMERO_DE_PRODUCOES;                              ANA0327
        INICIO_ARGUMENTO=ACUMULADO(PRODUCAO-1)+1;                      ANA0328
        FIM_ARGUMENTO=ACUMULADO(PRODUCAO);                             ANA0329
        TAMANHO=FIM_ARGUMENTO - INICIO_ARGUMENTO + 1;                  ANA0330
        ENTRADA(*)='1'B;                                               ANA0331
        CALL EMPILHAR_PRODUCOES_CONTINENTES;                           ANA0332
        IF EMPILHADAS_IGUAL>1                                          ANA0333
        THEN                                                           ANA0334
        DO;                                                            ANA0335
             INDICADOR_DE_PRECEDENCIA=INDICADOR_DE_PRECEDENCIA + 4;    ANA0336
             PRODUCAO=NUMERO_DE_PRODUCOES;                             ANA0337
        END;                                                           ANA0338
        ELSE IF EMPILHADAS>1                                           ANA0339
             THEN IF INDICADOR_DE_PRECEDENCIA=1                        ANA0340
                  THEN                                                 ANA0341
                  DO;                                                  ANA0342
                       CALL VERIFICAR_IGUAL_MENOR;                     ANA0343
                       IF INDICADOR_DE_PRECEDENCIA=3                   ANA0344
                       THEN PRODUCAO=NUMERO_DE_PRODUCOES;              ANA0345
                  END;                                                 ANA0346
                  ELSE;                                                ANA0347
             ELSE;                                                     ANA0348
    END;                                                               ANA0349
1/*****************************************************************/     ANA0350
 /*********** EMPILHAR PRODUCOES CONTINENTES (3.3.3.1) ************/     ANA0351
 /*****************************************************************/     ANA0352
    EMPILHAR_PRODUCOES_CONTINENTES:                                    ANA0353
    PROC;                                                              ANA0354
0        /*** ENCONTRAR AS PRODUCOES (3.3.3.1.1) ***/                 ANA0355
```

```
    DO J=INICIO_ARGUMENTO TO FIM_ARGUMENTO;                              ANA0356
         SAIDA(*)='0'B;                                                  ANA0357
         S=ARGUMENTO(J);                                                 ANA0358
         X=DOMINIO_OCORRE(S-1)+1;                                        ANA0359
         Y=DOMINIO_OCORRE(S);                                            ANA0360
         DO K=X TO Y;                                                    ANA0361
              SAIDA(CONTRADOMINIO_OCORRE(K))='1'B;                       ANA0362
         END;                                                            ANA0363
         ENTRADA=ENTRADA AND SAIDA;                                      ANA0364
    END;                                                                 ANA0365
0        /*** COMPARAR TAMANHO E ORDEM DOS SIMBOLOS (3.3.3.1.2) ***/    ANA0366
    EMPILHADAS,EMPILHADAS_IGUAL=0;                                       ANA0367
    DO J=1 TO NUMERO_DE_PRODUCOES;                                       ANA0368
         IF ENTRADA(J)='1'B                                              ANA0369
         THEN                                                            ANA0370
         DO;                                                             ANA0371
              CONTIDO=SIM;                                               ANA0372
              X=ACUMULADO(J-1);                                          ANA0373
              Y=ACUMULADO(J);                                            ANA0374
              T=Y-X;                                                     ANA0375
              IF T<TAMANHO                                               ANA0376
              THEN CONTIDO=NAO;                                          ANA0377
              ELSE                                                       ANA0378
              DO N=FIM_ARGUMENTO TO INICIO_ARGUMENTO BY -1;              ANA0379
                   IF ARGUMENTO(N) ¬= ARGUMENTO(Y)                       ANA0380
                   THEN CONTIDO=NAO;                                     ANA0381
                   Y=Y-1;                                                ANA0382
              END;                                                       ANA0383
              IF CONTIDO=SIM                                             ANA0384
              THEN                                                       ANA0385
              DO;                                                        ANA0386
                   IF T=TAMANHO                                          ANA0387
                   THEN EMPILHADAS_IGUAL=EMPILHADAS_IGUAL+1;             ANA0388
                   EMPILHADAS=EMPILHADAS+1;                              ANA0389
                   PILHA(EMPILHADAS)=J;                                  ANA0390
              END;                                                       ANA0391
         END;                                                            ANA0392
    END;                                                                 ANA0393
0   END;   /*** FIM DE EMPILHAR_PRODUCOES_CONTINENTES ***/               ANA0394
1/*************************************************************/         ANA0395
 /****************** VERIFICAR IGUAL MENOR (3.3.3.2) ******************/ ANA0396
 /*************************************************************/         ANA0397
    VERIFICAR_IGUAL_MENOR:                                               ANA0398
    PROC;                                                                ANA0399
    DO J=1 TO EMPILHADAS;                                                ANA0400
         IF PILHA(J)=PRODUCAO                                            ANA0401
         THEN;                                                           ANA0402
         ELSE                                                            ANA0403
         DO;                                                             ANA0404
              S=ACUMULADO(PILHA(J))-TAMANHO;                             ANA0405
              X=ARGUMENTO(S);                                            ANA0406
              I=DOMINIO_IGUAL(X-1)+1;                                    ANA0407
              T=DOMINIO_IGUAL(X);                                        ANA0408
              DO N=I TO T;                                               ANA0409
                   IF CONTRADOMINIO_IGUAL(N)=RAIZ(PRODUCAO)              ANA0410
                   THEN                                                  ANA0411
                   DO;                                                   ANA0412
                        N=T;                                             ANA0413
                        J=EMPILHADAS;                                    ANA0414
                        INDICADOR_DE_PRECEDENCIA=3;                      ANA0415
```

```
                    END;                                                    ANA0416
               END;                                                         ANA0417
               IF INDICADOR_DE_PRECEDENCIA=3                                ANA0418
               THEN RETURN;                                                 ANA0419
               I=DOMINIO_MENOR(X-1)+1;                                      ANA0420
               T=DOMINIO_MENOR(X);                                          ANA0421
               DO N=I TO T;                                                 ANA0422
                    IF CONTRADOMINIO_MENOR(N)=RAIZ(PRODUCAO)                ANA0423
                    THEN                                                    ANA0424
                    DO;                                                     ANA0425
                         N=T;                                               ANA0426
                         J=EMPILHADAS;                                      ANA0427
                         INDICADOR_DE_PRECEDENCIA=3;                        ANA0428
                    END;                                                    ANA0429
               END;                                                         ANA0430
          END;                                                              ANA0431
     END;                                                                   ANA0432
0    END;    /*** FIM DE VERIFICAR_IGUAL_MENOR ***/                        ANA0433
0    END;    /*** FIM DE COMPARAR UM ARGUMENTO COM OS DEMAIS ***/          ANA0434
0    END;    /*** FIM DE ANALISAR_ARGUMENTOS ***/                          ANA0435
1/*****************************************************************/       ANA0436
 /*************** GRAVAR CONTROLES (3.4) ***********************/         ANA0437
 /*****************************************************************/       ANA0438
     GRAVAR_CONTROLES:                                                      ANA0439
     PROC;                                                                  ANA0440
     DCL REGISTRO                          CHAR(80),                        ANA0441
         CARACTER(80)                      CHAR(1) DEF REGISTRO,            ANA0442
         CONTROLES(40)                     BIN FIXED BASED                  ANA0443
                                           (ADDR(REGISTRO)),                ANA0444
         ARQCTL2                           FILE RECORD OUTPUT;              ANA0445
0    IF ERRO=BRANCO                                                         ANA0446
     THEN CONTROLES(1)=0;                                                   ANA0447
     ELSE CARACTER(1)=ERRO;                                                 ANA0448
     CONTROLES(2)=NAOTERMINAIS;                                             ANA0449
     CONTROLES(3)=NUMERO_DE_SIMBOLOS;                                       ANA0450
     CONTROLES(4)=NUMERO_DE_PRODUCOES;                                      ANA0451
     CONTROLES(5)=TAMANHO_DOS_ARGUMENTOS;                                   ANA0452
     CONTROLES(6)=SIMBOLO_INICIAL;                                          ANA0453
     CONTROLES(7)=INDICADOR_DE_PRECEDENCIA;                                 ANA0454
     WRITE FILE (ARQCTL2) FROM (REGISTRO);                                  ANA0455
0    END;    /*** FIM DE GRAVAR_CONTROLES ***/                             ANA0456
-    END;    /*** FIM DO PROGRAMA ANAG103 ***/                             ANA0457
```

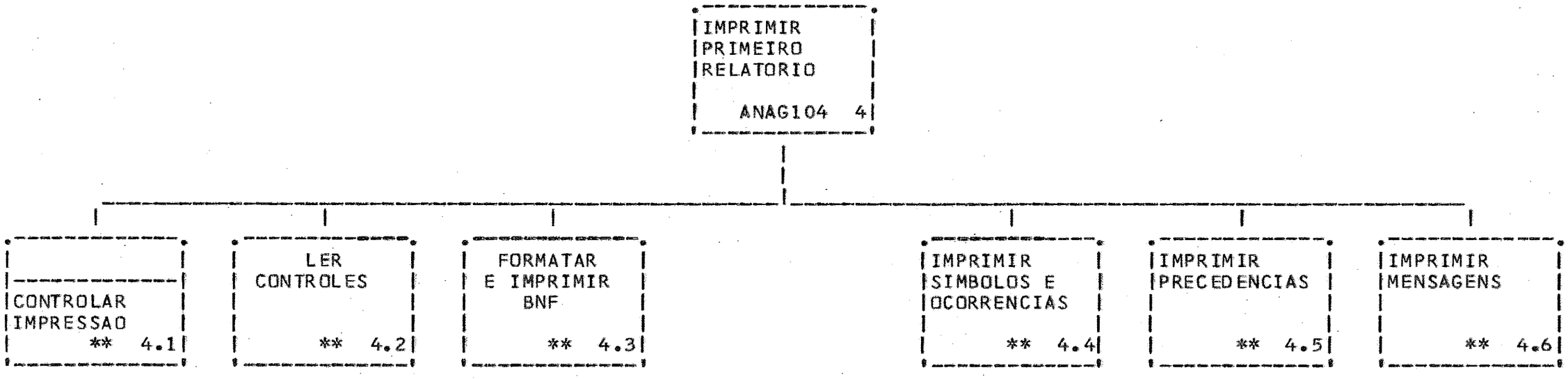

IMPRIMIR PRIMEIRO RELATORIO
ANAG104 4
CONTROLAR IMPRESSAO
** 4.1
LER CONTROLES
** 4.2
FORMATAR E IMPRIMIR BNF
** 4.3
IMPRIMIR SIMBOLOS E OCORRENCIAS
** 4.4
IMPRIMIR PRECEDENCIAS
** 4.5
IMPRIMIR MENSAGENS
** 4.6

FIGURA IV.13

```
                ****************************************
                *  LISTAGEM FONTE DO PROGRAMA ANAG104  *
                ****************************************

 /* IMPRIMIR PRIMEIRO RELATORIO (4)  VERSAO 15/05/79                    */ANA0001
0   ANAG104:                                                              ANA0002
    PROC OPTIONS(MAIN);                                                   ANA0003
0   DCL REGFONTE                          CHAR(80) BASED(P),              ANA0004
        PRODUCAO(80)                      CHAR(1) BASED(P),               ANA0005
        VETOR(40)                         BIN FIXED BASED(P),             ANA0006
        P                                 POINTER,                        ANA0007
        LINHA                             CHAR(100) VAR,                  ANA0008
        1 TABELA,                                                         ANA0009
          2 SIMB                          CHAR(67),                       ANA0010
          2 COD                           CHAR(2),                        ANA0011
          2 XXX                           CHAR(14) INIT(' '),             ANA0012
        SIMBOLO                           CHAR(32) VAR,                   ANA0013
        ORDEMCAB                          BIN FIXED INIT(1),              ANA0014
        DELIMITADOR                       CHAR(2) INIT('ZZ'),             ANA0015
        ERRO                              CHAR(1) INIT(' '),              ANA0016
        SIMBOLO_INICIAL                   BIN FIXED,                      ANA0017
        NUMERO_DE_SIMBOLOS                BIN FIXED,                      ANA0018
        NAOTERMINAIS                      BIN FIXED,                      ANA0019
        TAMANHO_DOS_ARGUMENTOS            BIN FIXED,                      ANA0020
        INDICADOR_DE_PRECEDENCIA          BIN FIXED,                      ANA0021
        NSIMBS                            BIN FIXED INIT(0),              ANA0022
        QUATRO                            CHAR(1) INIT('4'),              ANA0023
        CINCO                             CHAR(1) INIT('5'),              ANA0024
        BRANCO                            CHAR(1) INIT(' '),              ANA0025
        DOISBRANCOS                       CHAR(2) INIT('  '),             ANA0026
        PONT                              BIN FIXED,                      ANA0027
        PAGINA                            BIN FIXED INIT(0),              ANA0028
        TAMANHO                           BIN FIXED,                      ANA0029
        PRODUZ                            CHAR(7) INIT('  ::=  '),        ANA0030
        BARRA                             CHAR(7) INIT('    |  '),        ANA0031
        ANAGRAMA                          CHAR(22) INIT(                  ANA0032
                                         'A  N  A  G  R  A  M  A'),       ANA0033
        ANALISA                           CHAR(24) INIT(                  ANA0034
                                          'ANALISADOR DE GRAMATICAS'),    ANA0035
        DE_PRECEDENCIA                    CHAR(15) INIT(                  ANA0036
                                          ' DE PRECEDENCIA'),             ANA0037
        ANALISADOR                        CHAR(43),                       ANA0038
        PAG                               CHAR(5) INIT('PAG. ');          ANA0039
1   DCL ASTERISCO                         CHAR(116) VAR,                  ANA0040
        SEQUEN                            CHAR(14) INIT(                  ANA0041
                                          '* SEQUENCIAL *'),              ANA0042
        CABPRODUC                         CHAR(22) INIT(                  ANA0043
                                          'PRODUCOES DA GRAMATICA'),      ANA0044
        CABNAOTERM                        CHAR(42) INIT(                  ANA0045
                        '       RELACAO DOS NAO-TERMINAIS         *'),   ANA0046
        CABTERM                           CHAR(42) INIT(                  ANA0047
                        '         RELACAO DOS TERMINAIS           *'),   ANA0048
        OCORRENCIAS                       CHAR(31) INIT(                  ANA0049
                                        'PRODUCOES ONDE O SIMBOLO OCORRE'), ANA0050
        BRANCOS                           BIN FIXED,                      ANA0051
        RAIZANTERIOR                      BIN FIXED,                      ANA0052
        SEQUENCIA                         DEC FIXED(5,0) INIT(0),         ANA0053
        FIM                               BIN FIXED INIT(0),              ANA0054
       (K,                                                                ANA0055
        I,                                                                ANA0056
```

```
      J,                                                                ANA0057
      N)                                BIN FIXED,                      ANA0058
      TABELA_DE_SIMBOLOS(255)           CHAR(32),                       ANA0059
      PRIMEIRO_SIMBOLO                  CHAR(45) INIT(                  ANA0060
               '*                 SIMBOLO                      *'),     ANA0061
      SEGUNDO_SIMBOLO                   CHAR(44) INIT(                  ANA0062
               '              VIZINHANCA A DIREITA             *'),     ANA0063
      PRECEDENCIA                       CHAR(21) INIT(                  ANA0064
                                      '   PRECEDENCIA NO PAR'),         ANA0065
      CAB_MSG                           CHAR(32) INIT(                  ANA0066
                              'DIAGNOSTICOS E MENSAGENS DE ERRO'),      ANA0067
      DATE                              BUILTIN,                        ANA0068
      DATA_PLI                          CHAR(6),                        ANA0069
      ANO_MES_DIA(3)                    CHAR(2)                         ANA0070
                                        BASED(ADDR(DATA_PLI)),          ANA0071
      MES                               PIC '99',                       ANA0072
      DATA                              CHAR(9) VAR INIT(''),           ANA0073
      MESES_ANO(12)                     CHAR(3) INIT(                   ANA0074
                             'JAN','FEV','MAR','ABR','MAI','JUN',       ANA0075
                             'JUL','AGO','SET','OUT','NOV','DEZ'),      ANA0076
      TRACO                             CHAR(1) INIT('-'),              ANA0077
      RELATORIO                         CHAR(19) INIT(                  ANA0078
                                        'RELATORIO 01   -   '),         ANA0079
      RELATORIO_UM                      CHAR(28) VAR INIT('');          ANA0080
1   OPEN FILE(SYSPRINT) PAGESIZE(59);                                   ANA0081
0   ASTERISCO='';                                                       ANA0082
    DO K=1 TO 116;                                                      ANA0083
       ASTERISCO=ASTERISCO CAT '*';                                     ANA0084
    END;                                                                ANA0085
    ANALISADOR=ANALISA CAT DE_PRECEDENCIA;                              ANA0086
0   DATA_PLI=DATE;                                                      ANA0087
    MES=ANO_MES_DIA(2);                                                 ANA0088
    DATA=ANO_MES_DIA(3) CAT TRACO CAT MESES_ANO(MES)                    ANA0089
         CAT TRACO CAT ANO_MES_DIA(1);                                  ANA0090
    RELATORIO_UM=RELATORIO CAT DATA;                                    ANA0091
1/*****************************************************************/   ANA0092
 /***************** CONTROLAR IMPRESSAO (4.1) ********************/    ANA0093
 /*****************************************************************/   ANA0094
    ON ENDPAGE(SYSPRINT)                                                ANA0095
    BEGIN;                                                              ANA0096
         PAGINA=PAGINA+1;                                               ANA0097
         PUT EDIT(ANAGRAMA,PAG,PAGINA,ANALISADOR,RELATORIO_UM)          ANA0098
         (PAGE,SKIP,COL(50),A,COL(108),A,F(3),SKIP(2),COL(41),A,        ANA0099
         SKIP(2),COL(47),A);                                            ANA0100
         PUT EDIT(ASTERISCO)(SKIP(3),COL(5),A);                         ANA0101
         IF ORDEMCAB=1                                                  ANA0102
         THEN PUT EDIT(SEQUEN,CABPRODUC,'*')                            ANA0103
                 (COL(5),A,COL(37),A,COL(120),A);                       ANA0104
         IF ORDEMCAB=2                                                  ANA0105
         THEN PUT EDIT(SEQUEN,CABNAOTERM,OCORRENCIAS,'*')               ANA0106
                 (COL(5),A,A,COL(72),A,COL(120),A);                     ANA0107
         IF ORDEMCAB=3                                                  ANA0108
         THEN PUT EDIT(SEQUEN,CABTERM,OCORRENCIAS,'*')                  ANA0109
                 (COL(5),A,A,COL(72),A,COL(120),A);                     ANA0110
         IF ORDEMCAB=4                                                  ANA0111
         THEN PUT EDIT(PRIMEIRO_SIMBOLO,SEGUNDO_SIMBOLO,PRECEDENCIA,'*')ANA0112
                    (COL(5),A,A,A,COL(120),A);                          ANA0113
         IF ORDEMCAB=5                                                  ANA0114
         THEN PUT EDIT('*',CAB_MSG,'*')(COL(5),A,COL(45),A,COL(120),A); ANA0115
         PUT EDIT(ASTERISCO)(COL(5),A);                                 ANA0116
```

```
    END;                                                                    ANA0117
-   CALL LER_CONTROLES;                                                     ANA0118
0   CALL FORMATAR_E_IMPRIMIR_BNF;                                           ANA0119
0   IF ERRO=BRANCO OR ERRO=QUATRO OR ERRO=CINCO                             ANA0120
    THEN CALL IMPRIMIR_SIMBOLOS_E_OCORRENCIAS;                              ANA0121
0   IF ERRO=BRANCO                                                          ANA0122
    THEN CALL IMPRIMIR_PRECEDENCIAS;                                        ANA0123
0   CALL IMPRIMIR_MENSAGENS;                                                ANA0124
1/*************************************************************************/ ANA0125
 /***************** LER CONTROLES (4.2) ***************************/        ANA0126
 /*************************************************************************/ ANA0127
    LER_CONTROLES:                                                          ANA0128
    PROC;                                                                   ANA0129
    DCL ARQCTL2                              FILE RECORD INPUT,             ANA0130
        CONTROLES(40)                        BIN FIXED BASED(P),            ANA0131
        CARACTER(80)                         CHAR(1) BASED(P);              ANA0132
0   READ FILE (ARQCTL2) SET (P);                                            ANA0133
    NAOTERMINAIS=CONTROLES(2);                                              ANA0134
    NUMERO_DE_SIMBOLOS=CONTROLES(3);                                        ANA0135
    TAMANHO_DOS_ARGUMENTOS=CONTROLES(5);                                    ANA0136
    SIMBOLO_INICIAL=CONTROLES(6);                                           ANA0137
    INDICADOR_DE_PRECEDENCIA=CONTROLES(7);                                  ANA0138
0   IF CONTROLES(1)=0                                                       ANA0139
    THEN NUMERO_DE_SIMBOLOS = NUMERO_DE_SIMBOLOS - 1;                       ANA0140
    ELSE ERRO=CARACTER(1);                                                  ANA0141
0   END;    /*** FIM DE LER_CONTROLES ***/                                 ANA0142
1/*************************************************************************/ ANA0143
 /***************** FORMATAR E IMPRIMIR BNF (4.3) *****************/        ANA0144
 /*************************************************************************/ ANA0145
    FORMATAR_E_IMPRIMIR_BNF:                                                ANA0146
PROC;                                                                       ANA0147
DCL BNFONTE                                  FILE RECORD INPUT;             ANA0148
ON ENDFILE(BNFONTE)                                                         ANA0149
     FIM=1;                                                                 ANA0150
SIGNAL ENDPAGE(SYSPRINT);                                                   ANA0151
K=0;                                                                        ANA0152
DO WHILE(K=0);                                                              ANA0153
     READ FILE(BNFONTE) SET(P);                                             ANA0154
     IF PRODUCAO(1)=BRANCO                                                  ANA0155
     THEN;                                                                  ANA0156
     ELSE K=1;                                                              ANA0157
END;                                                                        ANA0158
DO WHILE(FIM=0);                                                            ANA0159
     LINHA='';                                                              ANA0160
     PONT=1;                                                                ANA0161
     CALL EXTRAIR_SIMBOLOS_E_BRANCOS;                                       ANA0162
     IF TAMANHO>0                                                           ANA0163
     THEN                                                                   ANA0164
     DO;                                                                    ANA0165
          IF PRODUCAO(1)='*'                                                ANA0166
          THEN                                                              ANA0167
          DO;                                                               ANA0168
               TAMANHO=0;                                                   ANA0169
               PONT=80;                                                     ANA0170
          END;                                                              ANA0171
          ELSE                                                              ANA0172
          DO;                                                               ANA0173
               LINHA=LINHA CAT SIMBOLO CAT PRODUZ;                          ANA0174
               RAIZANTERIOR=TAMANHO;                                        ANA0175
```

```
                SEQUENCIA=SEQUENCIA+1;                                  ANA0176
            END;                                                        ANA0177
        END;                                                            ANA0178
        ELSE IF PONT<80                                                 ANA0179
            THEN                                                        ANA0180
            DO;                                                         ANA0181
                DO K=1 TO RAIZANTERIOR;                                 ANA0182
                    LINHA=LINHA CAT BRANCO;                             ANA0183
                END;                                                    ANA0184
                LINHA=LINHA CAT BARRA;                                  ANA0185
                SEQUENCIA=SEQUENCIA+1;                                  ANA0186
            END;                                                        ANA0187
            ELSE;                                                       ANA0188
        DO WHILE(PONT<80);                                              ANA0189
            CALL EXTRAIR_SIMBOLOS_E_BRANCOS;                            ANA0190
            LINHA=LINHA CAT SIMBOLO CAT DOISBRANCOS;                    ANA0191
        END;                                                            ANA0192
        IF TAMANHO=0                                                    ANA0193
        THEN                                                            ANA0194
        DO;                                                             ANA0195
            PRODUCAO(1)=BRANCO;                                         ANA0196
            PUT EDIT(REGFONTE)(SKIP(2),COL(21),A);                      ANA0197
        END;                                                            ANA0198
        ELSE PUT EDIT(SEQUENCIA,LINHA)                                  ANA0199
                (SKIP(2),COL(10),P'999',COL(21),A);                     ANA0200
        READ FILE(BNFONTE) SET(P);                                      ANA0201
    END;                                                                ANA0202
    CLOSE FILE(BNFONTE);                                                ANA0203
1/*****************************************************************/ANA0204
 /**************** EXTRAIR SIMBOLOS E BRANCOS (4.3.1) ***************/ANA0205
 /*****************************************************************/ANA0206
    EXTRAIR_SIMBOLOS_E_BRANCOS:                                         ANA0207
    PROC;                                                               ANA0208
0   TAMANHO=0;                                                          ANA0209
    SIMBOLO='';                                                         ANA0210
0   IF PRODUCAO(PONT) NE BRANCO                                         ANA0211
    THEN                                                                ANA0212
    DO K=PONT TO 80;                                                    ANA0213
        IF PRODUCAO(K)=BRANCO                                           ANA0214
        THEN                                                            ANA0215
        DO;                                                             ANA0216
            PONT=K;                                                     ANA0217
            K=80;                                                       ANA0218
        END;                                                            ANA0219
        ELSE                                                            ANA0220
        DO;                                                             ANA0221
            TAMANHO=TAMANHO+1;                                          ANA0222
            IF TAMANHO<33                                               ANA0223
            THEN SIMBOLO=SIMBOLO CAT PRODUCAO(K);                       ANA0224
        END;                                                            ANA0225
    END;                                                                ANA0226
0   DO WHILE(PONT<81 AND PRODUCAO(PONT)=BRANCO);                        ANA0227
        PONT=PONT+1;                                                    ANA0228
    END;                                                                ANA0229
0   END;    /*** FIM DE EXTRAIR_SIMBOLOS_E_BRANCOS ***/                ANA0230
0   END;    /*** FIM DE FORMATAR_E_IMPRIMIR_BNF ***/                   ANA0231
1/*****************************************************************/ ANA0232
 /************** IMPRIMIR SIMBOLOS E OCORRENCIAS (4.4) *************/ ANA0233
 /*****************************************************************/ ANA0234
    IMPRIMIR_SIMBOLOS_E_OCORRENCIAS:                                    ANA0235
```

```
    PROC;                                                                ANA0236
0   DCL ARQTAB                          FILE RECORD INPUT,               ANA0237
        Q                               POINTER,                         ANA0238
        SIMBOLO_GRAVADO(2)              CHAR(32) BASED(Q),               ANA0239
        SIMBOLO_FONTE                   CHAR(32),                        ANA0240
        ARQOCOR                         FILE RECORD INPUT,               ANA0241
        TRESBRANCOS                     CHAR(3) INIT('   '),             ANA0242
        INICIO                          BIN FIXED INIT(1),               ANA0243
        FIM                             BIN FIXED,                       ANA0244
        MARGEM                          CHAR(3) VAR,                     ANA0245
        MSG01                           CHAR(49) INIT(                   ANA0246
              '*** SIMBOLO CRIADO COMO DELIMITADOR DE CADEIA ***'),      ANA0247
        X                               BIN FIXED INIT(2);               ANA0248
1   N=40;                                                                ANA0249
    SEQUENCIA=0;                                                         ANA0250
    FIM=NAOTERMINAIS;                                                    ANA0251
    CALL LER_TABELA_DE_SIMBOLOS;                                         ANA0252
    DO K=2 TO 3;                                                         ANA0253
         ORDEMCAB=K;                                                     ANA0254
         SIGNAL ENDPAGE(SYSPRINT);                                       ANA0255
         DO NSIMBS=INICIO TO FIM;                                        ANA0256
              CALL LER_TABELA_DE_SIMBOLOS;                               ANA0257
              TABELA_DE_SIMBOLOS(NSIMBS)=SIMBOLO_FONTE;                  ANA0258
              SEQUENCIA=SEQUENCIA+1;                                     ANA0259
              PUT EDIT(SEQUENCIA,SIMBOLO_FONTE)                          ANA0260
                   (SKIP(2),COL(10),P'999',COL(21),A);                   ANA0261
             J=0;                                                        ANA0262
             CALL LER_OCORRENCIAS;                                       ANA0263
             IF VETOR(N)>0                                               ANA0264
             THEN                                                        ANA0265
             DO;                                                         ANA0266
             IF VETOR(N)<10                                              ANA0267
             THEN MARGEM=BRANCO;                                         ANA0268
             ELSE IF VETOR(N)<100                                        ANA0269
                  THEN MARGEM=DOISBRANCOS;                               ANA0270
                  ELSE MARGEM=TRESBRANCOS;                               ANA0271
             PUT EDIT(MARGEM)(COL(60),A);                                ANA0272
             PUT EDIT(VETOR(N))(F(3));                                   ANA0273
             CALL LER_OCORRENCIAS;                                       ANA0274
        END;                                                             ANA0275
        DO WHILE(VETOR(N)>0);                                            ANA0276
             PUT EDIT(',')(A);                                           ANA0277
             J=J+1;                                                      ANA0278
             IF J=14                                                     ANA0279
             THEN                                                        ANA0280
             DO;                                                         ANA0281
                  IF VETOR(N)<10                                         ANA0282
                  THEN MARGEM=BRANCO;                                    ANA0283
                  ELSE IF VETOR(N)<100                                   ANA0284
                       THEN MARGEM=DOISBRANCOS;                          ANA0285
                       ELSE MARGEM=TRESBRANCOS;                          ANA0286
                  PUT EDIT(MARGEM)(SKIP,COL(60),A);                      ANA0287
                  J=0;                                                   ANA0288
             END;                                                        ANA0289
             PUT EDIT(VETOR(N))(F(3));                                   ANA0290
             CALL LER_OCORRENCIAS;                                       ANA0291
        END;                                                             ANA0292
END;                                                                     ANA0293
INICIO=NAOTERMINAIS+1;                                                   ANA0294
FIM=NUMERO_DE_SIMBOLOS;                                                  ANA0295
```

```
     END;                                                                    ANA0296
     NUMERO_DE_SIMBOLOS=NUMERO_DE_SIMBOLOS+1;                                ANA0297
     TABELA_DE_SIMBOLOS(NUMERO_DE_SIMBOLOS)=DELIMITADOR;                     ANA0298
     SEQUENCIA=SEQUENCIA+1;                                                  ANA0299
     PUT EDIT(SEQUENCIA,DELIMITADOR,MSG01)                                   ANA0300
     (SKIP(2),COL(10),P'999',COL(21),A,COL(65),A);                           ANA0301
     CLOSE FILE(ARQTAB);                                                     ANA0302
1/*****************************************************************/        ANA0303
 /***************** LER OCORRENCIAS (4.4.1) ***********************/        ANA0304
 /*****************************************************************/        ANA0305
     LER_OCORRENCIAS:                                                        ANA0306
     PROC;                                                                   ANA0307
0    N=N+1;                                                                  ANA0308
     IF N>40                                                                 ANA0309
     THEN                                                                    ANA0310
     DO;                                                                     ANA0311
          READ FILE (ARQOCOR) SET (P);                                       ANA0312
          N=1;                                                               ANA0313
     END;                                                                    ANA0314
0    END;    /*** FIM DE LER_OCORRENCIAS ***/                               ANA0315
1/*****************************************************************/        ANA0316
 /***************** LER TABELA DE SIMBOLOS (4.4.2) ****************/        ANA0317
 /*****************************************************************/        ANA0318
     LER_TABELA_DE_SIMBOLOS:                                                 ANA0319
     PROC;                                                                   ANA0320
     X=X+1;                                                                  ANA0321
     IF X>2                                                                  ANA0322
     THEN                                                                    ANA0323
     DO;                                                                     ANA0324
          READ FILE (ARQTAB) SET (Q);                                        ANA0325
          X=1;                                                               ANA0326
     END;                                                                    ANA0327
     SIMBOLO_FONTE=SIMBOLO_GRAVADO(X);                                       ANA0328
0    END;    /*** FIM DE LER_TABELA ***/                                    ANA0329
0    END;    /*** FIM DE IMPRIMIR_SIMBOLOS_E_OCORRENCIAS ***/               ANA0330
1/*****************************************************************/        ANA0331
 /******************* IMPRIMIR PRECEDENCIAS (4.5) *****************/        ANA0332
 /*****************************************************************/        ANA0333
     IMPRIMIR_PRECEDENCIAS:                                                  ANA0334
     PROC;                                                                   ANA0335
0    DCL ARQMAT                                FILE RECORD SEQUENTIAL INPUTANA0336
                                               BUFFERED ENV(REGIONAL(1)),  ANA0337
         LINHA_DA_MATRIZ(256)                  CHAR(1) BASED(P),           ANA0338
         K                                     BIN FIXED INIT(0),          ANA0339
         KCARAC(2)                             CHAR(1) BASED(ADDR(K)),     ANA0340
         ZERO                                  CHAR(1),                    ANA0341
         PRECED(7)                             CHAR(9) INIT(               ANA0342
         '<        ','=        ','>        ','<=     **','>=    ***',     ANA0343
         '><    ***','>=<   ***');                                          ANA0344
1    ORDEMCAB=4;                                                             ANA0345
     ZERO=KCARAC(2);                                                         ANA0346
     SIGNAL ENDPAGE(SYSPRINT);                                               ANA0347
     DO I=1 TO NUMERO_DE_SIMBOLOS;                                           ANA0348
          READ FILE (ARQMAT) SET (P);                                        ANA0349
          DO J=1 TO NUMERO_DE_SIMBOLOS;                                      ANA0350
               IF LINHA_DA_MATRIZ(J)=ZERO                                    ANA0351
               THEN;                                                         ANA0352
               ELSE                                                          ANA0353
               DO;                                                           ANA0354
                    KCARAC(2)=LINHA_DA_MATRIZ(J);                            ANA0355
```

```
                  PUT EDIT(I,TABELA_DE_SIMBOLOS(I),                          ANA0356
                        J,TABELA_DE_SIMBOLOS(J),PRECED(K))                   ANA0357
                        (SKIP(2),COL(9),F(4),COL(15),A,COL(53),              ANA0358
                        F(4),COL(59),A,COL(105),A);                          ANA0359
               END;                                                          ANA0360
          END;                                                               ANA0361
     END;                                                                    ANA0362
0    END;    /*** FIM DE IMPRIMIR_PRECEDENCIAS ***/                         ANA0363
1/*****************************************************************/         ANA0364
 /****************** IMPRIMIR MENSAGENS (4.6) ********************/         ANA0365
 /*****************************************************************/         ANA0366
     IMPRIMIR_MENSAGENS:                                                     ANA0367
     PROC;                                                                   ANA0368
0    DCL MSGERRO(5)                              CHAR(40) INIT(             ANA0369
      'NUMERO DE SIMBOLOS EXCEDE 254',                                       ANA0370
      'PRIMEIRA PRODUCAO TEM RAIZ EM BRANCO',                                ANA0371
      'EXISTE PRODUCAO FALTANDO ARGUMENTO',                                  ANA0372
      'GRAMATICA SEM SIMBOLO INICIAL',                                       ANA0373
      'GRAMATICA COM MAIS DE UM SIMBOLO INICIAL'),                           ANA0374
        AVISO                                     CHAR(40) INIT(             ANA0375
      'ERRO NA DEFINICAO DA BNF DA GRAMATICA  '),                            ANA0376
        SINAL_DE_ERRO                             CHAR(6) INIT('***** '),    ANA0377
        NUMERO_DO_ERRO                            PIC'9';                    ANA0378
0    DCL MENSAGEM(0:8)                           CHAR(63) INIT(             ANA0379
      'A GRAMATICA E'' DE PRECEDENCIA SIMPLES',                              ANA0380
      'A GRAMATICA E'' DE PRECEDENCIA FRACA',                                ANA0381
      'EXISTEM CONFLITOS ENVOLVENDO A RELACAO  '' > '' ',                    ANA0382
      'EXISTEM CONFLITOS DO TIPO  '' < = '' ',                               ANA0383
      'EXISTE MAIS DE UMA PRODUCAO COM MESMO ARGUMENTO',                     ANA0384
      'EXISTEM CONFLITOS DO TIPO  '' < = '' ',                               ANA0385
      'EXISTEM CONFLITOS ENVOLVENDO A RELACAO  '' > '' ',                    ANA0386
      'A GRAMATICA NAO E'' DE PRECEDENCIA',                                  ANA0387
      '(CONDICAO 2 DA DEFINICAO DE PRECEDENCIA FRACA NAO FOI ATENDIDA)');    ANA0388
0DCL NAOTERMINAL_INICIAL                          CHAR(34) INIT(             ANA0389
      ' - SIMBOLO INICIAL DA GRAMATICA:  '),                                 ANA0390
      ARGUMENTO_ACUMULADO                         CHAR(38) INIT(             ANA0391
      ' - TAMANHO ACUMULADO DOS ARGUMENTOS:  ');                             ANA0392
1    ORDEMCAB=5;                                                            ANA0393
     SIGNAL ENDPAGE(SYSPRINT);                                              ANA0394
     IF ERRO NE BRANCO                                                      ANA0395
     THEN                                                                   ANA0396
     DO;                                                                    ANA0397
          NUMERO_DO_ERRO=ERRO;                                              ANA0398
          PUT EDIT(SINAL_DE_ERRO,AVISO)(SKIP(5),COL(20),A,A);               ANA0399
          PUT EDIT(MSGERRO(NUMERO_DO_ERRO))(SKIP,COL(26),A);                ANA0400
     END;                                                                   ANA0401
     ELSE                                                                   ANA0402
     DO;                                                                    ANA0403
          IF INDICADOR_DE_PRECEDENCIA > 1                                   ANA0404
          THEN                                                              ANA0405
          DO;                                                               ANA0406
               PUT EDIT(SINAL_DE_ERRO,MENSAGEM(7))(SKIP(5),COL(20),A,A);    ANA0407
               PUT EDIT(MENSAGEM(INDICADOR_DE_PRECEDENCIA))                 ANA0408
               (SKIP,COL(26),A);                                            ANA0409
          END;                                                              ANA0410
          ELSE PUT EDIT(MENSAGEM(INDICADOR_DE_PRECEDENCIA),                 ANA0411
               NAOTERMINAL_INICIAL, TABELA_DE_SIMBOLOS(SIMBOLO_INICIAL),    ANA0412
               ARGUMENTO_ACUMULADO,TAMANHO_DOS_ARGUMENTOS)                  ANA0413
               (SKIP(5),COL(26),A,                                          ANA0414
                SKIP,COL(26),A,A,                                           ANA0415
```

```
            SKIP,COL(26),A,F(4));                                    ANA0416
         IF INDICADOR_DE_PRECEDENCIA=3                               ANA0417
         THEN PUT EDIT(MENSAGEM(8))(SKIP,COL(26),A);                 ANA0418
         IF INDICADOR_DE_PRECEDENCIA>4                               ANA0419
         THEN PUT EDIT(MENSAGEM(4))(SKIP,COL(26),A);                 ANA0420
   END;                                                              ANA0421
0  END;   /*** FIM DE IMPRIMIR_MENSAGENS ***/                        ANA0422
0  END;   /*** FIM DE ANAG104 ***/                                   ANA0423
```

As palavras não nascem amarradas,
elas saltam, se beijam, se dissolvem
no céu livre por vezes um desenho,
são puras, largas, autênticas, indevassáveis.

"Consideração do Poema"

CARLOS DRUMMOND DE ANDRADE

CAPÍTULO V

DETERMINAÇÃO DAS FUNÇÕES LINEARES DE PRECEDÊNCIA

Conhecidas as relações de precedência entre dois símbolos quaisquer de uma gramática, surge o problema de como representar estas relações para uso em um reconhecedor sintático. Entre as várias formas possíveis desta representação existe uma que utiliza dois vetores, de tal maneira que é possível reproduzir as relações de precedência IGUAL, MENOR ou MAIOR entre dois símbolos através da consulta a determinados elementos nestes vetores. O método distingue se a gramática é de precedência simples ou fraca e se a precedência é fraca, as relações IGUAL e MENOR são tratadas indistintamente. Os dois vetores são a representação de duas funções chamadas funções lineares de precedência.

A grande vantagem das funções de precedência é que elas são extremamente econômicas em termos de memória, além de possibilitarem um tempo de resposta às consultas bastante satisfatório. Em contrapartida, no entanto, este método tem três desvantagens: (a) nem toda gramática tem relações de precedência representáveis por funções lineares; (b) dados dois símbolos, as funções lineares, quando existem, são capazes de reproduzir exatamente a relação apenas se existir uma das três relações entre eles; dito de outra forma, quando não existe relação entre dois símbolos as funções lineares fornecem um resultado enganoso; (c) a recuperação de erro nos analisadores que usam funções de precedência não é muito eficiente.

O Módulo 02 do ANAGRAMA descreve e implementa um

método de determinação de funções lineares de precedência que é uma variante do método apresentado por AHO/ULLMAN[1].

V.1 - Definição das Entradas

Existem duas entradas para o método:

a) um conjunto de triplas (s_i, s_j, r), onde s_i, s_j representam símbolos de uma gramática e r é uma das três relações IGUAL, MENOR e MAIOR; não existe tripla quando entre dois símbolos não existe relação; se existe mais de uma relação entre dois símbolos, existirão tantas triplas quantas sejam estas relações;

b) uma indicação se a gramática é de precedência simples ou fraca.

V.2 - Definição da Saída

Uma indicação 'NÃO EXISTEM FUNÇÕES F e G', ou um par (F,G) de vetores de inteiros, tal que

(1) $F = (f_1, f_2 \ldots f_n)$;

(2) $G = (g_1, g_2 \ldots g_m)$;

(3) $f_i < g_j$ sempre que s_i MENOR s_j

$f_1 = g_2$ sempre que s_i IGUAL s_j

$f_i > g_j$ sempre que s_i MAIOR s_j

A função F tem n elementos, onde n é o número de símbolos de gramática. A função G tem m elementos, onde m é igual ao número de símbolos se a gramática é de precedência simples e é igual ao número de terminais se a gramática é de precedência fraca.

Estes vetores F e G são denominados de funções lineares de precedência para a gramática e elas podem ser usadas para representar as relações de precedências desta gramática.

V.3 - Descrição do Método Principal

Passo 1: Construir um grafo com n nós chamados de F_1, F_2 ... F_n e m nós chamados de G_1, G_2 ... G_m. Os nós F_i correspodem aos símbolos da gramática. Os nós G_j correspodem a todos os símbolos, se a gramática é de precedência simples e correspondem aos terminais, se ela é de precedência fraca.

Passo 2: Dada uma tripla (s_i, s_j, IGUAL) agrupar F_i e G_j em um mesmo nó. Este nó agora passa a agrupar todos os nós já previamente agrupados com F_i e G_j. Se a gramática é de precedência fraca, não executar este passo e tratar IGUAL como MENOR.

Passo 3: Dada uma tripla (s_i, s_j, MAIOR), criar uma ligação de F_i para G_j.
Dada uma tripla (s_i, s_j, MENOR), criar uma ligação de G_j para F_i.

Passo 4: Executados os Passos 2 e 3 para todas as triplas, verificar se o grafo resultante é cíclico (Método Secundário 1).

Passo 5: Se o grafo é cíclico não existem F e G.

Se o grafo é acíclico determinar os valores de F e G (Método Secundário 2).

V.3.1 - Método Secundário 1 - Verificação de Ciclos

Passo 1: Percorrer os nós procurando encontrar um sem descendentes.

Passo 2: Se foi encontrado um nó sem descendentes, removê-lo e novamente executar o passo 1.

Passo 3: Se o grafo resultante é vazio
então o grafo é acíclico
senão o grafo é cíclico.

V.3.2 - Método Secundário 2 - Gerar Valores das Funções Lineares

Passo 1: Rotular cada nó do grafo com um valor inicialmente 0.
Fazer n = 1

Passo 2: Se o n-ésimo nó tem descendentes
fazer rótulo do nó igual a m + 1, onde
m = máximo dos rótulos dos descendentes.

Passo 3: Se o nó não é o último
então fazer n = n + 1 e executar novamente o passo 2.

Passo 4: Se houver mudança de rótulo em algum nó
então fazer n = 1 e executar novamente o passo 2.

Passo 5: Fazer f_i = rótulo do nó onde F_i está agrupado e
g_j = rótulo do nó onde G_j está agrupado, para todos os f_i e g_j.

V.4 - Implementação do Módulo

O Módulo 02 é implementado como um programa de nome ANAG201. A representação dos grafos utilizados no método descrito é feita em termos de listas. Esta implementação utiliza recursos de memória alocados em função da gramática sendo analisada, de modo que problemas de estouro de memória devem ser analisados de acordo com cada instalação onde o Módulo estiver implantado. As figuras deste capítulo documentam o programa ANAG201, do qual é fornecida finalmente uma listagem em imagem de cartões.

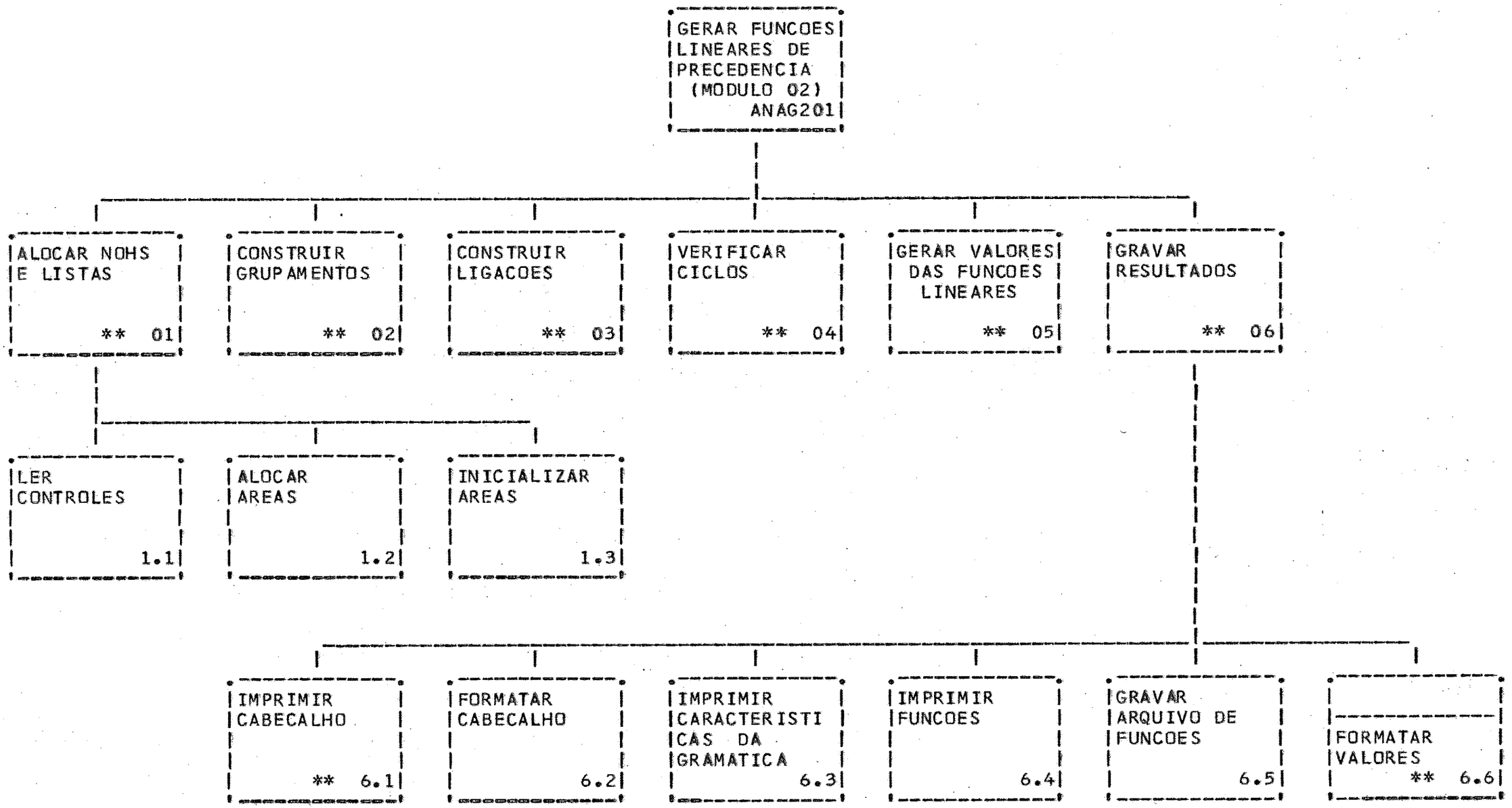

GERAR FUNCOES LINEARES DE PRECEDENCIA (MODULO 02) ANAG201
ALOCAR NOHS E LISTAS ** 01
CONSTRUIR GRUPAMENTOS ** 02
CONSTRUIR LIGACOES ** 03
VERIFICAR CICLOS ** 04
GERAR VALORES DAS FUNCOES LINEARES ** 05
GRAVAR RESULTADOS ** 06
LER CONTROLES 1.1
ALOCAR AREAS 1.2
INICIALIZAR AREAS 1.3
IMPRIMIR CABECALHO ** 6.1
FORMATAR CABECALHO 6.2
IMPRIMIR CARACTERISTICAS DA GRAMATICA 6.3
IMPRIMIR FUNCOES 6.4
GRAVAR ARQUIVO DE FUNCOES 6.5
FORMATAR VALORES ** 6.6

FIGURA V.1

ANAGRAMA - MODULO 02 - GERAR FUNCOES LINEAR DE PRECEDENCIA

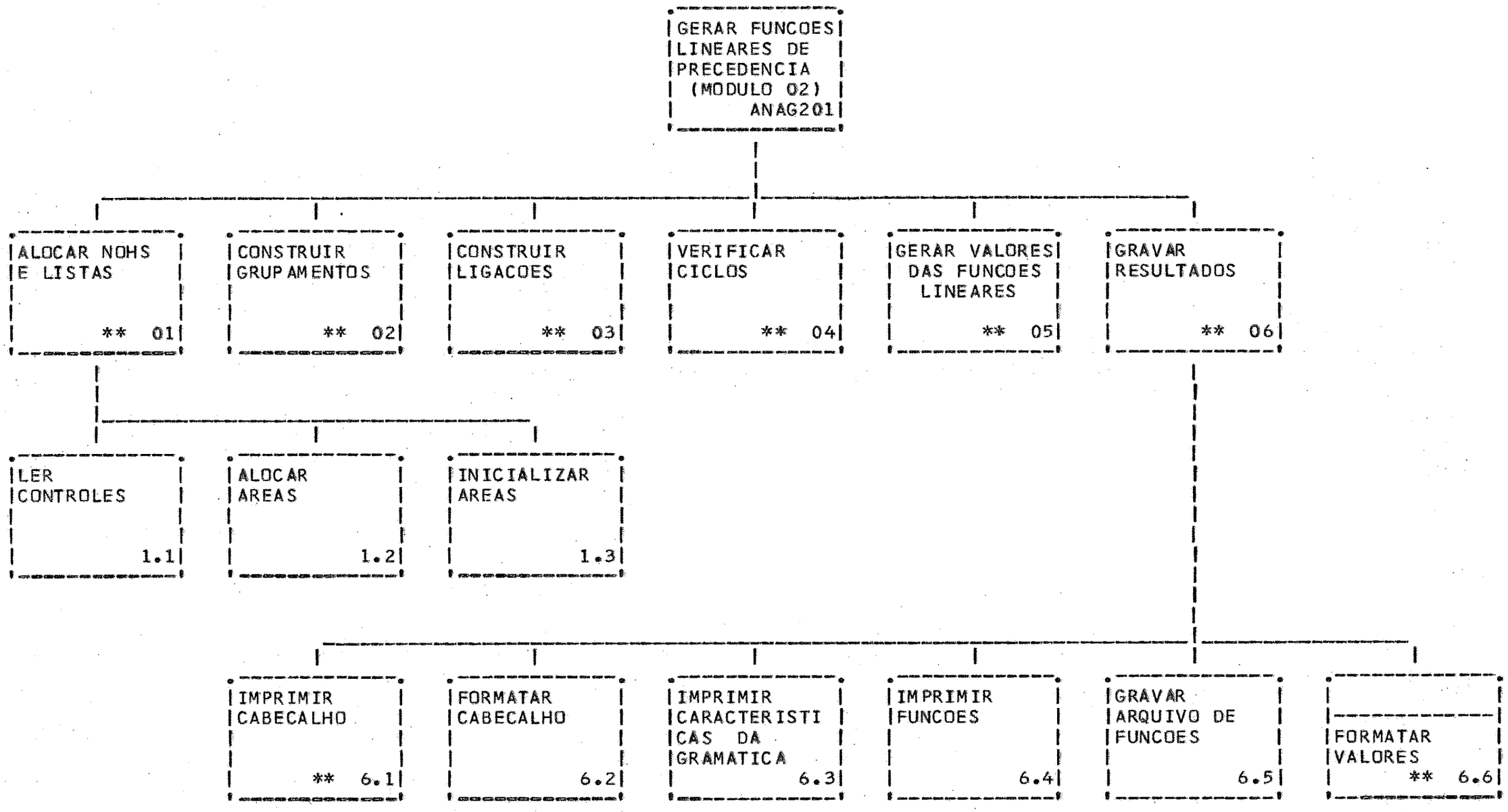


FIGURA V.1

ANAGRAMA - MODULO 02 - CONSTRUIR GRUPAMENTOS (2)

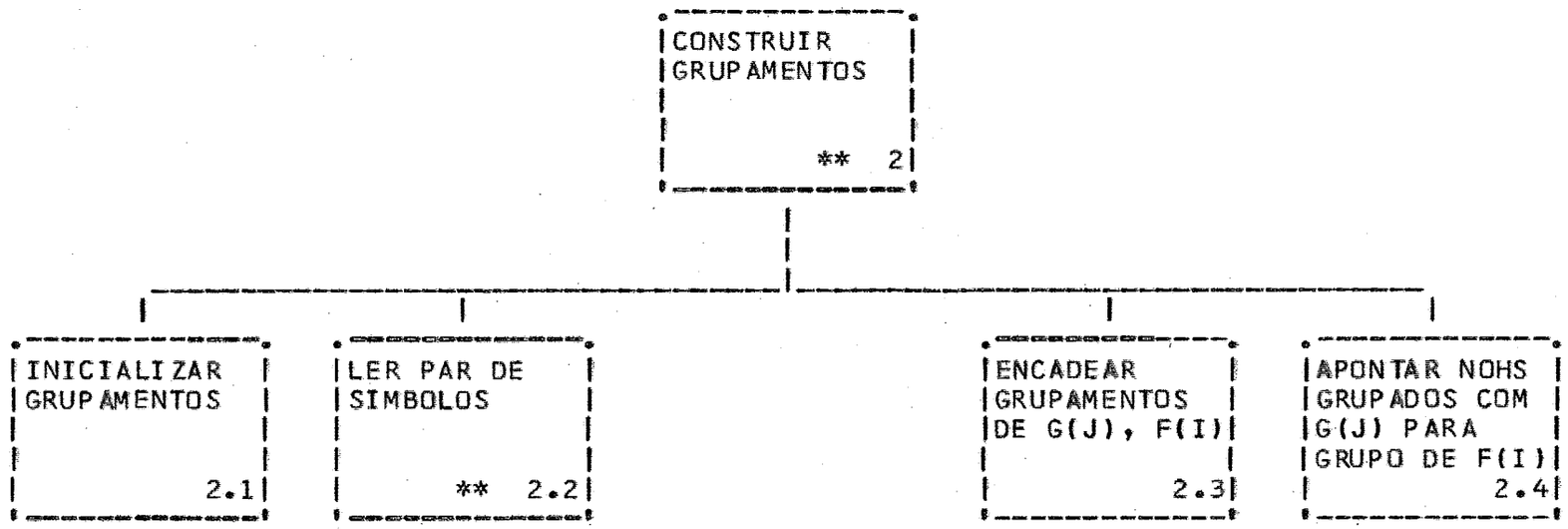


FIGURA V.2

ANAGRAMA - MODULO 02 - CONSTRUIR LIGACOES (3)

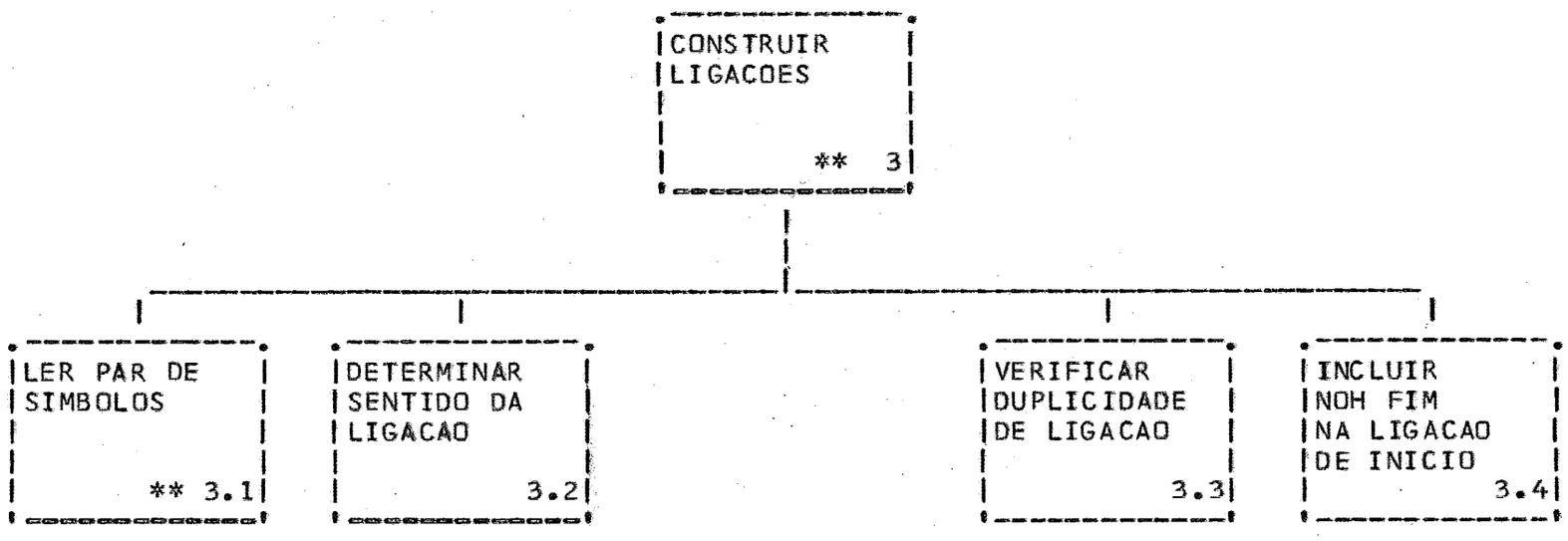


FIGURA V.3

ANAGRAMA - MODULO 02 - VERIFICAR CICLOS (4)

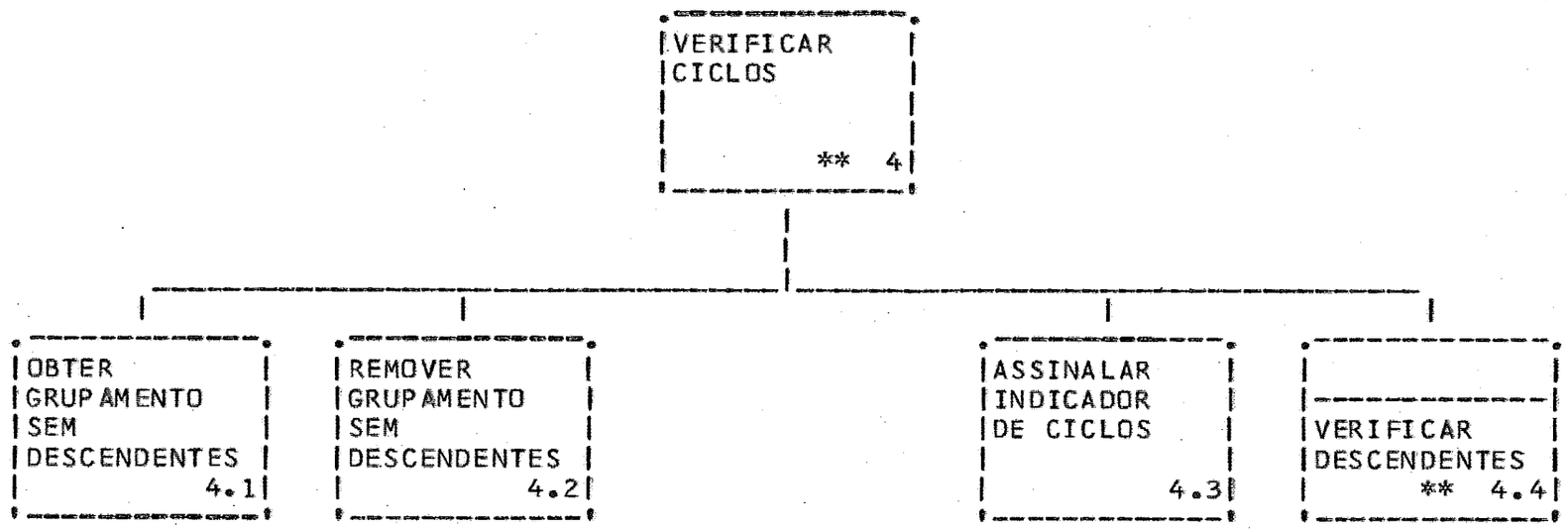


FIGURA V.4

ANAGRAMA - MODULO 02 - GERAR FUNCOES LINEARES (5)

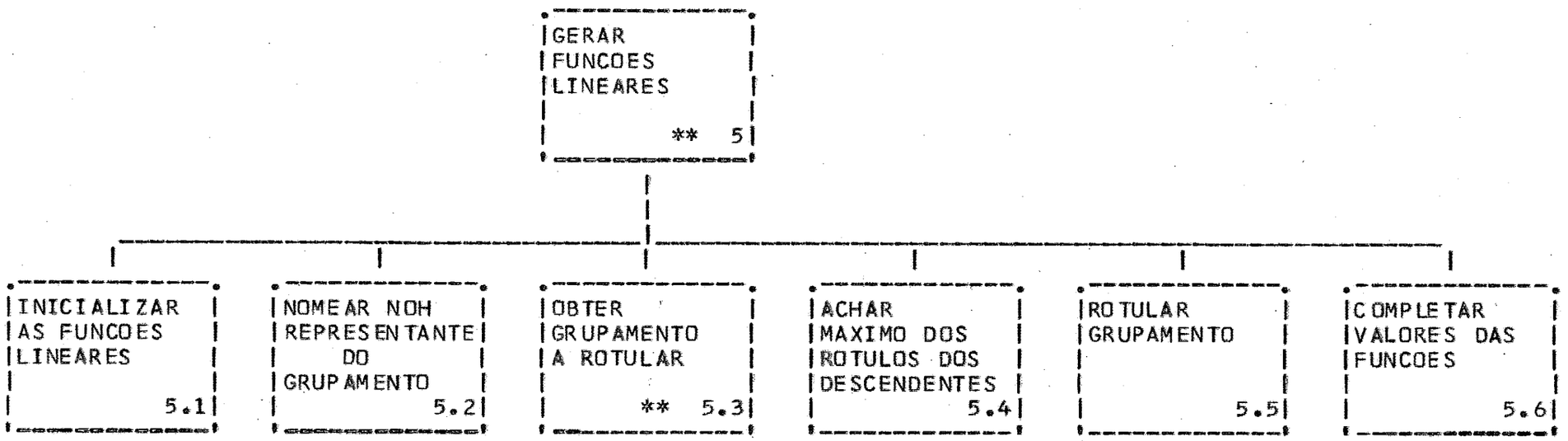


FIGURA V.5

ANAGRAMA - MODULO 02 - GERAR FUNCOES LINEARES DE PRECEDENCIA

ENTRADA

ARQCTL2

INDICADOR DE PRECEDENCIA

NAO TERMINAIS

NUMERO DE SIMBOLOS

ARQREL

PROCESSO

01 ALOCAR NOHS E LISTAS.

NOHS_F TEM 'NUMERO DE SIMBOLOS' NOHS. NOHS_G TEM 'NUMERO DE SIMBOLOS' NOHS SE A GRAMATICA E' DE PRECEDENCIA E TEM 'NUMERO DE TERMINAIS' NOHS SE A GRAMATICA E' DE PRECEDENCIA FRACA. GRUPAMENTOS TEM A SOMA DE NOHS ALOCADOS PARA 'F' E 'G'. LIGACOES TEM NUMERO DE NOHS IGUAL AS 'MAIOR' E 'MENOR' SOMADAS OU SEJA 8(NUMERO DE PRODUCOES + TAMANHO DOS ARGUMENTOS). ESTAS AREAS SAO INICIALIZADAS.

02 CONSTRUIR GRUPAMENTOS.

PARA CADA NOHS_F(I) E NOHS G(J) CONSTRUIR UMA LISTA CONTENDO APENAS UM ELEMENTO CUJA INFORMA CAO E' ESTE NOHS_F(I) OU NOHS_G(J). ESTAS LISTAS SAO OS GRUPAMENTOS DOS NOHS_F E NOHS_G. SE A PRECEDECIA E' SIMPLES OBTER TODAS AS TRIPLAS (S(I), S(J), IGUAL). PARA CADA TRIPLA CONCATENAR O GRUPAMENTO DE NOHS_G(J) COM O GUPAMENTO DE NOHS_F(I), OBTENDO NOVA LISTA QUE SERA' GRUPAMENTO DE TODOS OS NOHS_F E NOHS_G DAS LISTAS CONCATENADAS.

03 CONSTRUIR LIGACOES.

PARA CADA NOHS F(I) E NOHS G(J) ASSOCIAR UMA LISTA INICIALMENTE VAZIA. ESTAS LISTAS SAO AS LIGACOES DOS NOHS_F E NOHS_G. EM SEGUIDA OBTER TODAS AS TRIPLAS DO TIPO (S(I), S(J), MENOR) E (S(I), S(J), MAIOR). SE A RELACAO E' 'MAIOR' CRIAR UM ELEMENTO EM LIGACOES DE NOHS_F(I) CUJA INFORMACAO E' O GRUPAMENTO DE NOHS_G(J). SE A LIGACAO E' 'MENOR' UM ELEMENTO EM LIGACOES DE NOHS_G(J) CUJA INFORMACAO E' O GRUPAMENTO DO NOHS_F(I)

SAIDA

NUMERO DE SIMBOLOS

NAO TERMINAIS

LIGACOES

INDICADOR DE PRECEDENCIA

NOHS F

NOHS G

GRUPAMENTOS

CONTROLE

NOHS F

NOHS G

GRUPAMENTOS

LIGACOES

INDICADOR D
PRECEDENCIA

NOHS F
NOHS G

GRUPAMENTOS

LIGACOES

NUMERO DE
SIMBOLOS
NAO
TERMINAIS

INDICADOR D
PRECEDENCIA

INDICADOR D
CICLOS

CONTROLE

NAO
TERMINAIS

FUNCAO F
FUNCAO G

04 VERIFICAR CICLOS

05 GERAR VALORES DAS FUNCOES LINEARES

06 GRAVAR RESULTADOS.

AS FUNCOES 'F' E 'G', SE EXISTEM, SAO GRAVADOS EM DISCO JUNTO COM OS CONTROLES. O RELA TORIO INDICA SE HA' FUNCOES LINEARES E, SE HOUVER, LISTA OS SEUS VALORES.

INDICADOR D
CICLOS

FUNCAO F
FUNCAO G

RELATORIO

ARQFUNC

FIGURA V.6

ANAGRAMA - MODULO 02 - VERIFICAR CICLOS (4)

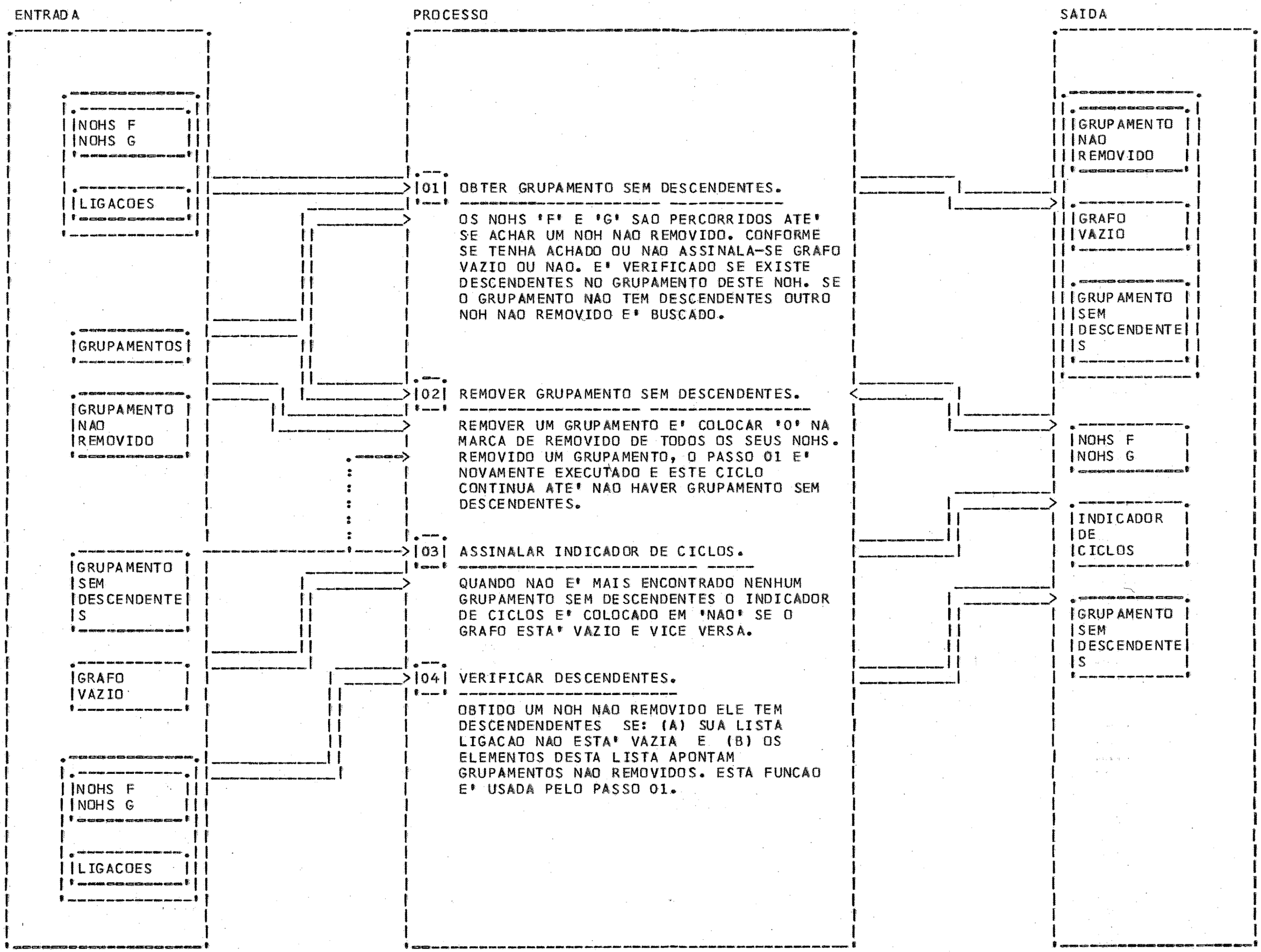


FIGURA V.7

ANAGRAMA - MODULO 02 - GERAR VALORES DAS FUNCOES LINEARES (5)

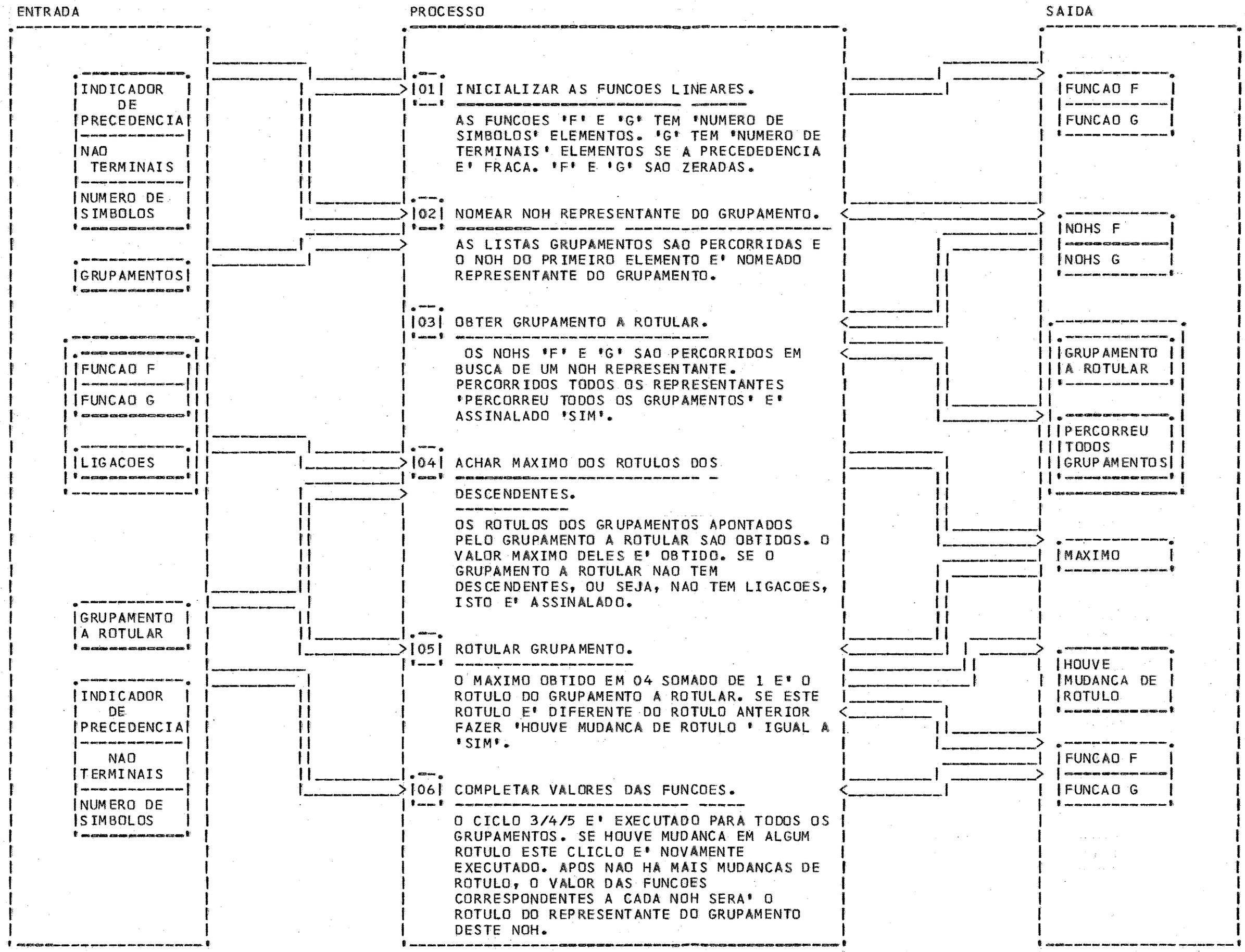


FIGURA V.8

```
      ****************************************
      *  LISTAGEM FONTE DO PROGRAMA ANAG201  *
      ****************************************


 /* GERAR FUNCOES LINEARES DE PRECEDENCIA - VERSAO 26/04/79          */ANA0001
-ANAG201:                                                              ANA0002
 PROC OPTIONS (MAIN);                                                  ANA0003
-/* DECLARACAO DAS VARIAVEIS INTERNAS                                 */ANA0004
ODCL F_MARCA_DE_REMOVIDO(1)              CHAR(1) CONTROLLED,           ANA0005
     F_PONTEIRO_AO_GRUPAMENTO(1)         BIN FIXED CONTROLLED,         ANA0006
     F_PONTEIRO_A_LIGACOES(1)            BIN FIXED CONTROLLED;         ANA0007
ODCL G_MARCA_DE_REMOVIDO(1:1)            CHAR(1) CONTROLLED,           ANA0008
     G_PONTEIRO_AO_GRUPAMENTO(1:1)       BIN FIXED CONTROLLED,         ANA0009
     G_PONTEIRO_A_LIGACOES(1:1)          BIN FIXED CONTROLLED;         ANA0010
ODCL 1 GRUPAMENTO(1)                     CONTROLLED,                   ANA0011
       2 TIPO_DE_NOH                     CHAR(1),                      ANA0012
       2 NOH                             BIN FIXED,                    ANA0013
       2 PONTEIRO                        BIN FIXED;                    ANA0014
ODCL 1 LIGACOES(1)                       CONTROLLED,                   ANA0015
       2 GRUPAMENTO_APONTADO             BIN FIXED,                    ANA0016
       2 PONTEIRO                        BIN FIXED,                    ANA0017
     CONTROLE                            CHAR(80),                     ANA0018
     NUMERO_DE_SIMBOLOS                  BIN FIXED,                    ANA0019
     NAOTERMINAIS                        BIN FIXED,                    ANA0020
     INDICADOR_DE_PRECEDENCIA            BIN FIXED,                    ANA0021
     INDICADOR_DE_CICLOS                 CHAR(1);                      ANA0022
ODCL 1 PAR_RELACAO(13)                   BASED(P),                     ANA0023
       2 A                               BIN FIXED,                    ANA0024
       2 B                               BIN FIXED,                    ANA0025
       2 C                               BIN FIXED,                    ANA0026
     P                                   POINTER,                      ANA0027
     ARQREL                              FILE RECORD INPUT;            ANA0028
ODCL FUNCAO_F(1)                         BIN FIXED CONTROLLED,         ANA0029
     FUNCAO_G(1)                         BIN FIXED CONTROLLED;         ANA0030
1/* DECLARACAO DAS CONSTANTES E FUNCOES                               */ANA0031
ODCL NAO                                 CHAR(1) INIT('N');            ANA0032
-CALL ALOCAR_NOHS_E_LISTAS;                                            ANA0033
 CALL CONSTRUIR_GRUPAMENTOS;                                           ANA0034
 CALL CONSTRUIR_LIGACOES;                                              ANA0035
 CALL VERIFICAR_CICLOS;                                                ANA0036
 IF INDICADOR_DE_CICLOS = NAO                                          ANA0037
 THEN CALL GERAR_VALORES_FUNCOES_LINEARES;                             ANA0038
 ELSE;                                                                 ANA0039
 CALL GRAVAR_RESULTADOS;                                               ANA0040
1/*****************************************************************/ANA0041
 /***************** ALOCAR NOHS E LISTAS (1) **************************/ANA0042
 /*****************************************************************/ANA0043
OALOCAR_NOHS_E_LISTAS:                                                 ANA0044
 PROC;                                                                 ANA0045
-DCL NUMERO_DE_TERMINAIS                  BIN FIXED,                    ANA0046
     NUMERO_DE_PRODUCOES                 BIN FIXED,                    ANA0047
     TAMANHO_DOS_ARGUMENTOS              BIN FIXED,                    ANA0048
     (I, K)                              BIN FIXED,                    ANA0049
```

```
     PRECEDENCIA_FRACA                  BIN FIXED INIT(1),                 ANA0050
     UM                                 CHAR(1) INIT('1'),                 ANA0051
     BRANCO                             CHAR(1) INIT(' '),                 ANA0052
     ADDR                               BUILTIN,                           ANA0053
     CONTROLES(40)                      BIN FIXED BASED(ADDR(CONTROLE)),   ANA0054
     ARQCTL2                            FILE RECORD INPUT;                 ANA0055
-/* LER CONTROLES (1.1)                                                  */ANA0056
OREAD FILE (ARQCTL2) INTO (CONTROLE);                                     ANA0057
 NAOTERMINAIS = CONTROLES(2);                                             ANA0058
 NUMERO_DE_SIMBOLOS = CONTROLES(3);                                       ANA0059
 NUMERO_DE_PRODUCOES = CONTROLES(4);                                      ANA0060
 TAMANHO_DOS_ARGUMENTOS = CONTROLES(5);                                   ANA0061
 INDICADOR_DE_PRECEDENCIA = CONTROLES(7);                                 ANA0062
 NUMERO_DE_TERMINAIS = NUMERO_DE_SIMBOLOS - NAOTERMINAIS;                 ANA0063
-/* ALOCAR AREAS (1.2)                                                   */ANA0064
OALLOCATE F_MARCA_DE_REMOVIDO(NUMERO_DE_SIMBOLOS);                        ANA0065
 ALLOCATE F_PONTEIRO_AO_GRUPAMENTO(NUMERO_DE_SIMBOLOS);                   ANA0066
 ALLOCATE F_PONTEIRO_A_LIGACOES(NUMERO_DE_SIMBOLOS);                      ANA0067
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                          ANA0068
 THEN                                                                     ANA0069
 DO;                                                                      ANA0070
      K = NAOTERMINAIS + 1;                                               ANA0071
      ALLOCATE G_MARCA_DE_REMOVIDO(K : NUMERO_DE_SIMBOLOS);               ANA0072
      ALLOCATE G_PONTEIRO_AO_GRUPAMENTO(K : NUMERO_DE_SIMBOLOS);          ANA0073
      ALLOCATE G_PONTEIRO_A_LIGACOES(K : NUMERO_DE_SIMBOLOS);             ANA0074
      K = NUMERO_DE_TERMINAIS;                                            ANA0075
 END;                                                                     ANA0076
 ELSE                                                                     ANA0077
 DO;                                                                      ANA0078
      K = NUMERO_DE_SIMBOLOS;                                             ANA0079
      ALLOCATE G_MARCA_DE_REMOVIDO(1:NUMERO_DE_SIMBOLOS);                 ANA0080
      ALLOCATE G_PONTEIRO_AO_GRUPAMENTO(1:NUMERO_DE_SIMBOLOS);            ANA0081
      ALLOCATE G_PONTEIRO_A_LIGACOES(1:NUMERO_DE_SIMBOLOS);               ANA0082
 END;                                                                     ANA0083
 ALLOCATE GRUPAMENTO(K + NUMERO_DE_SIMBOLOS);                             ANA0084
 ALLOCATE LIGACOES(8 * (NUMERO_DE_PRODUCOES + TAMANHO_DOS_ARGUMENTOS));   ANA0085
1/* INICIALIZAR AREAS (1.3)                                              */ANA0086
OF_PONTEIRO_AO_GRUPAMENTO, F_PONTEIRO_A_LIGACOES,                         ANA0087
 G_PONTEIRO_AO_GRUPAMENTO, G_PONTEIRO_A_LIGACOES,                         ANA0088
 NOH, GRUPAMENTO.PONTEIRO,                                                ANA0089
 GRUPAMENTO_APONTADO, LIGACOES.PONTEIRO = 0;                              ANA0090
 DO K = 1 TO NUMERO_DE_SIMBOLOS;                                          ANA0091
      F_MARCA_DE_REMOVIDO(K) = UM;                                        ANA0092
 END;                                                                     ANA0093
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                          ANA0094
 THEN I = NAOTERMINAIS + 1;                                               ANA0095
 ELSE I = 1;                                                              ANA0096
 DO K = I TO NUMERO_DE_SIMBOLOS;                                          ANA0097
      G_MARCA_DE_REMOVIDO(K) = UM;                                        ANA0098
 END;                                                                     ANA0099
 TIPO_DE_NOH = BRANCO;                                                    ANA0100
OEND;   /*** FIM DE ALOCAR_NOHS_E_LISTAS ***/                            ANA0101
1/*****************************************************************/     ANA0102
 /****************** CONSTRUIR GRUPAMENTOS (2) ********************/     ANA0103
 /*****************************************************************/     ANA0104
```

```
OCONSTRUIR_GRUPAMENTOS:                                                    ANA0105
 PROC;                                                                     ANA0106
-DCL PRECEDENCIA_FRACA                  BIN FIXED INIT(1),                 ANA0107
     PRIMEIRO_NOH_F                     BIN FIXED,                         ANA0108
     PRIMEIRO_NOH_G                     BIN FIXED,                         ANA0109
     ULTIMO_NOH_F                       BIN FIXED,                         ANA0110
     SIMBOLO_F                          BIN FIXED,                         ANA0111
     SIMBOLO_G                          BIN FIXED,                         ANA0112
     IGUAL                              BIN FIXED INIT(2),                 ANA0113
     PROCURA_IGUAL                      CHAR(1),                           ANA0114
     SIM                                CHAR(1) INIT('S'),                 ANA0115
     NAO                                CHAR(1) INIT('N'),                 ANA0116
     F                                  CHAR(1) INIT('F'),                 ANA0117
     G                                  CHAR(1) INIT('G'),                 ANA0118
     J                                  BIN FIXED INIT(13),                ANA0119
     (I, N, K)                          BIN FIXED;                         ANA0120
-/* INICIALIZAR GRUPAMENTOS (2.1)                                         */ANA0121
0DO K = 1 TO NUMERO_DE_SIMBOLOS;                                           ANA0122
      F_PONTEIRO_AO_GRUPAMENTO(K) = K;                                     ANA0123
      TIPO_DE_NOH(K) = F;                                                  ANA0124
      NOH(K) = K;                                                          ANA0125
 END;                                                                      ANA0126
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                           ANA0127
 THEN N = NAOTERMINAIS + 1;                                                ANA0128
 ELSE N = 1;                                                               ANA0129
 I = NUMERO_DE_SIMBOLOS;                                                   ANA0130
 DO K = N TO NUMERO_DE_SIMBOLOS;                                           ANA0131
      I = I + 1;                                                           ANA0132
      G_PONTEIRO_AO_GRUPAMENTO(K) = I;                                     ANA0133
      TIPO_DE_NOH(I) = G;                                                  ANA0134
      NOH(I) = K;                                                          ANA0135
 END;                                                                      ANA0136
1IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                           ANA0137
 THEN RETURN;                                                              ANA0138
 ELSE;                                                                     ANA0139
 CALL LER_PAR_DE_SIMBOLOS;                                                 ANA0140
 DO WHILE (SIMBOLO_F > 0);                                                 ANA0141
-/* ENCADEAR GRUPAMENTOS DE G(J) E F(I) (2.3)                             */ANA0142
0     I, PRIMEIRO_NOH_F = F_PONTEIRO_AO_GRUPAMENTO(SIMBOLO_F);             ANA0143
      DO WHILE (I > 0);                                                    ANA0144
           ULTIMO_NOH_F = I;                                               ANA0145
           I = GRUPAMENTO.PONTEIRO(I);                                     ANA0146
      END;                                                                 ANA0147
      PRIMEIRO_NOH_G = G_PONTEIRO_AO_GRUPAMENTO(SIMBOLO_G);                ANA0148
      IF PRIMEIRO_NOH_G = PRIMEIRO_NOH_F                                   ANA0149
      THEN;                                                                ANA0150
      ELSE GRUPAMENTO.PONTEIRO(ULTIMO_NOH_F) = PRIMEIRO_NOH_G;             ANA0151
-/* APONTAR NOHS GRUPADOS COM G(J) PARA GRUPAMENTO DE F(I) (2.4)          */ANA0152
0     I=PRIMEIRO_NOH_G;                                                    ANA0153
      DO WHILE (I > 0);                                                    ANA0154
           IF TIPO_DE_NOH(I) = F                                           ANA0155
           THEN F_PONTEIRO_AO_GRUPAMENTO(NOH(I)) = PRIMEIRO_NOH_F;         ANA0156
           ELSE G_PONTEIRO_AO_GRUPAMENTO(NOH(I)) = PRIMEIRO_NOH_F;         ANA0157
           I = GRUPAMENTO.PONTEIRO(I);                                     ANA0158
      END;                                                                 ANA0159
```

```
-/* LER PAR DE SIMBOLOS (2.2)                                          */ANA0160
0     CALL LER_PAR_DE_SIMBOLOS;                                          ANA0161
0END;                                                                    ANA0162
0CLOSE FILE (ARQREL);                                                    ANA0163
1/*****************************************************************/ANA0164
 /*************** LER PAR DE SIMBOLOS (2.2) ***********************/ANA0165
 /*****************************************************************/ANA0166
0LER_PAR_DE_SIMBOLOS:                                                    ANA0167
 PROC;                                                                   ANA0168
0PROCURA_IGUAL = SIM;                                                    ANA0169
 DO WHILE (PROCURA_IGUAL = SIM);                                         ANA0170
      J = J + 1;                                                         ANA0171
      IF J > 13                                                          ANA0172
      THEN                                                               ANA0173
      DO;                                                                ANA0174
           READ FILE (ARQREL) SET (P);                                   ANA0175
           J = 1;                                                        ANA0176
      END;                                                               ANA0177
      ELSE;                                                              ANA0178
      IF C(J) = IGUAL                                                    ANA0179
      THEN                                                               ANA0180
      DO;                                                                ANA0181
           SIMBOLO_F = A(J);                                             ANA0182
           SIMBOLO_G = B(J);                                             ANA0183
           PROCURA_IGUAL = NAO;                                          ANA0184
      END;                                                               ANA0185
      ELSE;                                                              ANA0186
      IF A(J) = 0                                                        ANA0187
      THEN                                                               ANA0188
      DO;                                                                ANA0189
           SIMBOLO_F = 0;                                                ANA0190
           PROCURA_IGUAL = NAO;                                          ANA0191
      END;                                                               ANA0192
      ELSE;                                                              ANA0193
0END;                                                                    ANA0194
0END;    /*** FIM DE LER_PAR_DE_SIMBOLOS ***/                           ANA0195
0END;    /*** FIM DE CONSTRUIR_GRUPAMENTOS ***/                         ANA0196
1/*****************************************************************/ANA0197
 /****************** CONSTRUIR LIGACOES (3) ***********************/ANA0198
 /*****************************************************************/ANA0199
0CONSTRUIR_LIGACOES:                                                     ANA0200
 PROC;                                                                   ANA0201
-/* DECLARACAO DAS VARIAVEIS INTERNAS                                  */ANA0202
0DCL  SIMBOLO_F                        BIN FIXED,                        ANA0203
      SIMBOLO_G                        BIN FIXED,                        ANA0204
      RELACAO                          BIN FIXED,                        ANA0205
      1 INICIO,                                                          ANA0206
        2 NOH                          BIN FIXED,                        ANA0207
        2 TIPO                         CHAR(1),                          ANA0208
      FIM                              BIN FIXED,                        ANA0209
      INDICADOR_DE_DUPLICIDADE         CHAR(1);                          ANA0210
-/* DECLARACAO DAS VARIAVEIS DE TRABALHO                               */ANA0211
0DCL  PRECEDENCIA_FRACA                BIN FIXED INIT(1),                ANA0212
      ULTIMO_NOH_DA_LIGACAO            BIN FIXED,                        ANA0213
      GRUPAMENTO_A_SER_APONTADO        BIN FIXED,                        ANA0214
```

```
     PROCURA_PAR                        CHAR(1),                          ANA0215
     DISPONIVEL                         BIN FIXED INIT(0),                ANA0216
     J                                  BIN FIXED INIT(13),               ANA0217
     (I, K)                             BIN FIXED;                        ANA0218
-/* DECLARACAO DAS CONSTANTES E FUNCOES                                  */ANA0219
0DCL MAIOR                              BIN FIXED INIT(3),                ANA0220
     MENOR                              BIN FIXED INIT(1),                ANA0221
     IGUAL                              BIN FIXED INIT(2),                ANA0222
     SIM                                CHAR(1) INIT('S'),                ANA0223
     NAO                                CHAR(1) INIT('N'),                ANA0224
     F                                  CHAR(1) INIT('F'),                ANA0225
     G                                  CHAR(1) INIT('G');                ANA0226
1CALL LER_PAR_DE_SIMBOLOS;                                               ANA0227
 DO WHILE (SIMBOLO_F > 0);                                               ANA0228
-/* DETERMINAR SENTIDO DA LIGACAO (3.2)                                  */ANA0229
0      IF RELACAO = MAIOR                                                ANA0230
       THEN                                                              ANA0231
       DO;                                                               ANA0232
            INICIO.TIPO = F;                                             ANA0233
            INICIO.NOH = SIMBOLO_F;                                      ANA0234
            FIM = SIMBOLO_G;                                             ANA0235
       END;                                                              ANA0236
       ELSE;                                                             ANA0237
       IF RELACAO = MENOR                                                ANA0238
       THEN                                                              ANA0239
       DO;                                                               ANA0240
            INICIO.TIPO = G;                                             ANA0241
            INICIO.NOH = SIMBOLO_G;                                      ANA0242
            FIM = SIMBOLO_F;                                             ANA0243
       END;                                                              ANA0244
       ELSE;                                                             ANA0245
       IF RELACAO = IGUAL                                                ANA0246
       THEN IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA              ANA0247
            THEN                                                         ANA0248
            DO;                                                          ANA0249
                 INICIO.TIPO = G;                                        ANA0250
                 INICIO.NOH = SIMBOLO_G;                                 ANA0251
                 FIM = SIMBOLO_F;                                        ANA0252
            END;                                                         ANA0253
            ELSE;                                                        ANA0254
       ELSE;                                                             ANA0255
1/* VERIFICAR DUPLICIDADE (3.3)                                         */ANA0256
0      INDICADOR_DE_DUPLICIDADE = NAO;                                   ANA0257
       IF INICIO.TIPO = F                                                ANA0258
       THEN I = F_PONTEIRO_A_LIGACOES(INICIO.NOH);                       ANA0259
       ELSE I = G_PONTEIRO_A_LIGACOES(INICIO.NOH);                       ANA0260
       DO WHILE (I > 0);                                                 ANA0261
            K = GRUPAMENTO_APONTADO(I);                                  ANA0262
            DO WHILE (K > 0);                                            ANA0263
                 IF GRUPAMENTO.NOH(K) = FIM  AND                         ANA0264
                 TIPO_DE_NOH(K) NE INICIO.TIPO                           ANA0265
                 THEN INDICADOR_DE_DUPLICIDADE = SIM;                    ANA0266
                 ELSE;                                                   ANA0267
                 K = GRUPAMENTO.PONTEIRO(K);                             ANA0268
            END;                                                         ANA0269
```

```
          I = LIGACOES.PONTEIRO(I);                                    ANA0270
          IF INDICADOR_DE_DUPLICIDADE = SIM                            ANA0271
          THEN I = 0;                                                  ANA0272
          ELSE;                                                        ANA0273
     END;                                                              ANA0274
1/* INCLUIR NOH FIM NA LIGACAO DE INICIO (3.4)                       */ANA0275
0    IF INDICADOR_DE_DUPLICIDADE = SIM                                 ANA0276
     THEN;                                                             ANA0277
     ELSE                                                              ANA0278
     DO;                                                               ANA0279
          IF INICIO.TIPO = F                                           ANA0280
          THEN                                                         ANA0281
          DO;                                                          ANA0282
               I = F_PONTEIRO_A_LIGACOES(INICIO.NOH);                  ANA0283
               K = F_PONTEIRO_AO_GRUPAMENTO(INICIO.NOH);               ANA0284
          END;                                                         ANA0285
          ELSE                                                         ANA0286
          DO;                                                          ANA0287
               I = G_PONTEIRO_A_LIGACOES(INICIO.NOH);                  ANA0288
               K = G_PONTEIRO_AO_GRUPAMENTO(INICIO.NOH);               ANA0289
          END;                                                         ANA0290
          ULTIMO_NOH_DA_LIGACAO = 0;                                   ANA0291
          DO WHILE (I > 0);                                            ANA0292
               ULTIMO_NOH_DA_LIGACAO = I;                              ANA0293
               I = LIGACOES.PONTEIRO(I);                               ANA0294
          END;                                                         ANA0295
          IF INICIO.TIPO = F                                           ANA0296
          THEN GRUPAMENTO_A_SER_APONTADO =                             ANA0297
               G_PONTEIRO_AO_GRUPAMENTO(FIM);                          ANA0298
          ELSE GRUPAMENTO_A_SER_APONTADO =                             ANA0299
               F_PONTEIRO_AO_GRUPAMENTO(FIM);                          ANA0300
          DISPONIVEL = DISPONIVEL + 1;                                 ANA0301
          IF ULTIMO_NOH_DA_LIGACAO = 0                                 ANA0302
          THEN                                                         ANA0303
          DO WHILE (K > 0);                                            ANA0304
               I = GRUPAMENTO.NOH(K);                                  ANA0305
               IF TIPO_DE_NOH(K) = F                                   ANA0306
               THEN F_PONTEIRO_A_LIGACOES(I) = DISPONIVEL;             ANA0307
               ELSE G_PONTEIRO_A_LIGACOES(I) = DISPONIVEL;             ANA0308
               K = GRUPAMENTO.PONTEIRO(K);                             ANA0309
          END;                                                         ANA0310
          ELSE LIGACOES.PONTEIRO(ULTIMO_NOH_DA_LIGACAO) = DISPONIVEL;  ANA0311
          GRUPAMENTO_APONTADO(DISPONIVEL) = GRUPAMENTO_A_SER_APONTADO; ANA0312
     END;                                                              ANA0313
-/* LER PAR DE SIMBOLOS (3.1)                                        */ANA0314
0    CALL LER_PAR_DE_SIMBOLOS;                                         ANA0315
0END;   /* FIM DO WHILE (SIMBOLO_F > 0)  */                           ANA0316
1/*****************************************************************/ANA0317
 /***************** LER PAR DE SIMBOLOS (3.1) **********************/ANA0318
 /*****************************************************************/ANA0319
0LER_PAR_DE_SIMBOLOS:                                                 ANA0320
 PROC;                                                               ANA0321
0PROCURA_PAR = SIM;                                                  ANA0322
 DO WHILE (PROCURA_PAR = SIM);                                       ANA0323
     J = J + 1;                                                      ANA0324
```

```
     IF J > 13                                                          ANA0325
     THEN                                                               ANA0326
     DO;                                                                ANA0327
          READ FILE (ARQREL) SET (P);                                   ANA0328
          J = 1;                                                        ANA0329
     END;                                                               ANA0330
     ELSE;                                                              ANA0331
     SIMBOLO_F = A(J);                                                  ANA0332
     SIMBOLO_G = B(J);                                                  ANA0333
     RELACAO = C(J);                                                    ANA0334
     IF SIMBOLO_F = 0                                                   ANA0335
     THEN PROCURA_PAR = NAO;                                            ANA0336
     ELSE IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA               ANA0337
          THEN IF SIMBOLO_G > NAOTERMINAIS                              ANA0338
               THEN PROCURA_PAR = NAO;                                  ANA0339
               ELSE;                                                    ANA0340
          ELSE PROCURA_PAR = NAO;                                       ANA0341
OEND;                                                                   ANA0342
OEND;   /*** FIM DE LER_PAR_DE_SIMBOLOS ***/                           ANA0343
OEND;   /*** FIM DE CONSTRUIR_LIGACOES ***/                           ANA0344
1/*****************************************************************/ANA0345
 /******************* VERIFICAR CICLOS (4) ************************/ANA0346
 /*****************************************************************/ANA0347
OVERIFICAR_CICLOS:                                                      ANA0348
 PROC;                                                                  ANA0349
-/* DECLARACAO DAS VARIAVEIS INTERNAS                                 */ANA0350
ODCL GRUPAMENTO_NAO_REMOVIDO          BIN FIXED,                        ANA0351
     GRAFO_VAZIO                      CHAR(1) INIT('N'),                ANA0352
     GRUPAMENTO_SEM_DESCENDENTES      CHAR(1) INIT('S');                ANA0353
-/* DECLARACAO DAS VARIAVEIS DE TRABALHO                              */ANA0354
ODCL MARCA                            CHAR(1),                          ANA0355
     TIPO_NAO_REMOVIDO                CHAR(1),                          ANA0356
     INICIO_G                         BIN FIXED,                        ANA0357
     (I, J, K)                        BIN FIXED;                        ANA0358
-/* DECLARACAO DAS CONSTANTES E FUNCOES                               */ANA0359
ODCL PRECEDENCIA_FRACA                BIN FIXED INIT(1),                ANA0360
     NAO_REMOVIDO                     CHAR(1) INIT('1'),                ANA0361
     REMOVIDO                         CHAR(1) INIT('0'),                ANA0362
     F                                CHAR(1) INIT('F'),                ANA0363
     G                                CHAR(1) INIT('G'),                ANA0364
     SIM                              CHAR(1) INIT('S'),                ANA0365
     NAO                              CHAR(1) INIT('N');                ANA0366
1IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0367
 THEN INICIO_G = NAOTERMINAIS + 1;                                      ANA0368
 ELSE INICIO_G = 1;                                                     ANA0369
 DO WHILE(GRUPAMENTO_SEM_DESCENDENTES = SIM);                           ANA0370
-/* OBTER GRUPAMENTO SEM DESCENDENTES (4.1)                           */ANA0371
0     GRUPAMENTO_NAO_REMOVIDO = 0;                                      ANA0372
      GRAFO_VAZIO = SIM;                                                ANA0373
      DO K = 1 TO NUMERO_DE_SIMBOLOS;                                   ANA0374
           IF F_MARCA_DE_REMOVIDO(K) = NAO_REMOVIDO                     ANA0375
           THEN                                                         ANA0376
           DO;                                                          ANA0377
                GRUPAMENTO_NAO_REMOVIDO = K;                            ANA0378
                TIPO_NAO_REMOVIDO = F;                                  ANA0379
```

```
          GRAFO_VAZIO = NAO;                                          ANA0380
          I = F_PONTEIRO_A_LIGACOES(K);                               ANA0381
          CALL VERIFICAR_DESCENDENTES;                                ANA0382
          IF GRUPAMENTO_SEM_DESCENDENTES = SIM                        ANA0383
          THEN K = NUMERO_DE_SIMBOLOS;                                ANA0384
          ELSE;                                                       ANA0385
     END;                                                             ANA0386
     ELSE;                                                            ANA0387
END;                                                                  ANA0388
IF GRUPAMENTO_NAO_REMOVIDO = 0  OR                                    ANA0389
GRUPAMENTO_SEM_DESCENDENTES = NAO                                     ANA0390
THEN                                                                  ANA0391
DO K = INICIO_G TO NUMERO_DE_SIMBOLOS;                                ANA0392
     IF G_MARCA_DE_REMOVIDO(K) = NAO_REMOVIDO                         ANA0393
     THEN                                                             ANA0394
     DO;                                                              ANA0395
          GRUPAMENTO_NAO_REMOVIDO = K;                                ANA0396
          TIPO_NAO_REMOVIDO = G;                                      ANA0397
          GRAFO_VAZIO = NAO;                                          ANA0398
          I = G_PONTEIRO_A_LIGACOES(K);                               ANA0399
          CALL VERIFICAR_DESCENDENTES;                                ANA0400
          IF GRUPAMENTO_SEM_DESCENDENTES = SIM                        ANA0401
          THEN K = NUMERO_DE_SIMBOLOS;                                ANA0402
          ELSE;                                                       ANA0403
     END;                                                             ANA0404
     ELSE;                                                            ANA0405
END;                                                                  ANA0406
ELSE;                                                                 ANA0407
IF GRUPAMENTO_NAO_REMOVIDO = 0                                        ANA0408
     THEN GRUPAMENTO_SEM_DESCENDENTES = NAO;                          ANA0409
     ELSE;                                                            ANA0410
1/* REMOVER GRUPAMENTO SEM DESCENDENTES (4.2)                       */ANA0411
0    IF GRUPAMENTO_SEM_DESCENDENTES = SIM                             ANA0412
     THEN                                                             ANA0413
     DO;                                                              ANA0414
          IF TIPO_NAO_REMOVIDO = F                                    ANA0415
          THEN I = F_PONTEIRO_AO_GRUPAMENTO(GRUPAMENTO_NAO_REMOVIDO); ANA0416
          ELSE I = G_PONTEIRO_AO_GRUPAMENTO(GRUPAMENTO_NAO_REMOVIDO); ANA0417
          DO WHILE (I > 0);                                           ANA0418
               IF TIPO_DE_NOH(I) = F                                  ANA0419
               THEN F_MARCA_DE_REMOVIDO(NOH(I)) = REMOVIDO;           ANA0420
               ELSE G_MARCA_DE_REMOVIDO(NOH(I)) = REMOVIDO;           ANA0421
               I = GRUPAMENTO.PONTEIRO(I);                            ANA0422
          END;                                                        ANA0423
     END;                                                             ANA0424
     ELSE;                                                            ANA0425
0END;   /* FIM DO WHILE GRUPAMENTO_SEM_DESCENDENTES = SIM */          ANA0426
-/* ASSINALAR INDICADOR DE CICLOS (4.3)                             */ANA0427
0IF GRAFO_VAZIO = SIM                                                 ANA0428
 THEN INDICADOR_DE_CICLOS = NAO;                                      ANA0429
 ELSE INDICADOR_DE_CICLOS = SIM;                                      ANA0430
1/*****************************************************************/ANA0431
 /***************** VERIFICAR DESCENDENTES (4.4) ********************/ANA0432
 /*****************************************************************/ANA0433
0VERIFICAR_DESCENDENTES:                                              ANA0434
```

```
 PROC;                                                                  ANA0435
OGRUPAMENTO_SEM_DESCENDENTES = SIM;                                     ANA0436
 DO WHILE(I > 0);                                                       ANA0437
      J = GRUPAMENTO_APONTADO(I);                                       ANA0438
      IF TIPO_DE_NOH(J) = F                                             ANA0439
      THEN MARCA = F_MARCA_DE_REMOVIDO(NOH(J));                         ANA0440
      ELSE MARCA = G_MARCA_DE_REMOVIDO(NOH(J));                         ANA0441
      IF MARCA = NAO_REMOVIDO                                           ANA0442
      THEN                                                              ANA0443
      DO;                                                               ANA0444
           GRUPAMENTO_SEM_DESCENDENTES = NAO;                           ANA0445
           I = 0;                                                       ANA0446
      END;                                                              ANA0447
      ELSE I = LIGACOES.PONTEIRO(I);                                    ANA0448
 END;                                                                   ANA0449
OEND;    /* FIM DE VERIFICAR_DESCENDENTES */                            ANA0450
OEND;    /*** FIM DE VERIFICAR_CICLOS ***/                             ANA0451
1/*****************************************************************/   ANA0452
 /***************** GERAR VALORES DAS FUNCOES LINEARES (5) ************/ ANA0453
 /*****************************************************************/   ANA0454
OGERAR_VALORES_FUNCOES_LINEARES:                                        ANA0455
 PROC;                                                                  ANA0456
-/* DECLARACAO DAS VARIAVEIS INTERNAS                                 */ ANA0457
ODCL HOUVE_MUDANCA_DE_ROTULO          CHAR(1) INIT('S'),                ANA0458
     PERCORREU_TODOS_GRUPAMENTOS      CHAR(1),                          ANA0459
     MAXIMO                           BIN FIXED,                        ANA0460
     1 GRUPAMENTO_A_ROTULAR,                                            ANA0461
       2 NOH                          BIN FIXED,                        ANA0462
       2 PONTEIRO_LIGACOES            BIN FIXED,                        ANA0463
       2 TIPO                         CHAR(1);                          ANA0464
-/* DECLARACAO DAS VARIAVEIS DE TRABALHO                              */ ANA0465
ODCL INICIO_G                         BIN FIXED,                        ANA0466
     F_INICIO_PROCURA                 BIN FIXED,                        ANA0467
     G_INICIO_PROCURA                 BIN FIXED,                        ANA0468
     FUNCAO_DO_REPRESENTANTE          BIN FIXED,                        ANA0469
     (I, J, K)                        BIN FIXED;                        ANA0470
-/* DECLARACAO DAS CONSTANTES E FUNCOES                               */ ANA0471
ODCL PRECEDENCIA_FRACA                BIN FIXED INIT(1),                ANA0472
     REPRESENTANTE                    CHAR(1) INIT('R'),                ANA0473
     SIM                              CHAR(1) INIT('S'),                ANA0474
     NAO                              CHAR(1) INIT('N'),                ANA0475
     F                                CHAR(1) INIT ('F'),               ANA0476
     G                                CHAR(1) INIT ('G');               ANA0477
1/* INICIALIZAR AS FUNCOES LINEARES (5.1)                             */ ANA0478
OALLOCATE FUNCAO_F(NUMERO_DE_SIMBOLOS);                                 ANA0479
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0480
 THEN ALLOCATE FUNCAO_G(NAOTERMINAIS + 1: NUMERO_DE_SIMBOLOS);          ANA0481
 ELSE ALLOCATE FUNCAO_G(NUMERO_DE_SIMBOLOS);                            ANA0482
 FUNCAO_F = 0;                                                          ANA0483
 FUNCAO_G = 0;                                                          ANA0484
-/* NOMEAR NOH REPRESENTANTE DO GRUPAMENTO (5.2)                      */ ANA0485
OIF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0486
 THEN INICIO_G = NAOTERMINAIS + 1;                                      ANA0487
 ELSE INICIO_G = 1;                                                     ANA0488
 DO K = 1 TO NUMERO_DE_SIMBOLOS;                                        ANA0489
```

```
      I = F_PONTEIRO_AO_GRUPAMENTO(K);                                    ANA0490
      IF TIPO_DE_NOH(I) = F                                               ANA0491
      THEN F_MARCA_DE_REMOVIDO(GRUPAMENTO.NOH(I)) = REPRESENTANTE;        ANA0492
      ELSE G_MARCA_DE_REMOVIDO(GRUPAMENTO.NOH(I)) = REPRESENTANTE;        ANA0493
 END;                                                                     ANA0494
 DO K = INICIO_G TO NUMERO_DE_SIMBOLOS;                                   ANA0495
      I = G_PONTEIRO_AO_GRUPAMENTO(K);                                    ANA0496
      IF TIPO_DE_NOH(I) = F                                               ANA0497
      THEN F_MARCA_DE_REMOVIDO(GRUPAMENTO.NOH(I)) = REPRESENTANTE;        ANA0498
      ELSE G_MARCA_DE_REMOVIDO(GRUPAMENTO.NOH(I)) = REPRESENTANTE;        ANA0499
 END;                                                                     ANA0500
100 WHILE (HOUVE_MUDANCA_DE_ROTULO = SIM);                               ANA0501
      GRUPAMENTO_A_ROTULAR.TIPO = F;                                      ANA0502
      GRUPAMENTO_A_ROTULAR.NOH = 0;                                       ANA0503
      G_INICIO_PROCURA = INICIO_G;                                        ANA0504
      PERCORREU_TODOS_GRUPAMENTOS, HOUVE_MUDANCA_DE_ROTULO = NAO;         ANA0505
      CALL OBTER_GRUPAMENTO_A_ROTULAR;                                    ANA0506
0     DO WHILE(PERCORREU_TODOS_GRUPAMENTOS = NAO);                        ANA0507
0/* ACHAR MAXIMO DOS ROTULOS DOS DESCENDENTES (5.4)                      */ANA0508
0          MAXIMO = -1;                                                  ANA0509
           I = GRUPAMENTO_A_ROTULAR.PONTEIRO_LIGACOES;                   ANA0510
           DO WHILE (I > 0);                                             ANA0511
                K = GRUPAMENTO_APONTADO(I);                              ANA0512
                IF TIPO_DE_NOH(K) = F                                    ANA0513
                THEN J = FUNCAO_F(GRUPAMENTO.NOH(K));                    ANA0514
                ELSE J = FUNCAO_G(GRUPAMENTO.NOH(K));                    ANA0515
                IF J > MAXIMO                                            ANA0516
                THEN MAXIMO = J;                                         ANA0517
                ELSE;                                                    ANA0518
                I = LIGACOES.PONTEIRO(I);                                ANA0519
           END;                                                          ANA0520
0/* ROTULAR GRUPAMENTO (5.5)                                            */ANA0521
0          K = MAXIMO + 1;                                               ANA0522
           IF MAXIMO = -1                                                ANA0523
           THEN;                                                         ANA0524
           ELSE                                                          ANA0525
           DO;                                                           ANA0526
                IF GRUPAMENTO_A_ROTULAR.TIPO = F                         ANA0527
                THEN                                                     ANA0528
                DO;                                                      ANA0529
                     J = FUNCAO_F(GRUPAMENTO_A_ROTULAR.NOH);             ANA0530
                     FUNCAO_F(GRUPAMENTO_A_ROTULAR.NOH) = K;             ANA0531
                END;                                                     ANA0532
                ELSE                                                     ANA0533
                DO;                                                      ANA0534
                     J = FUNCAO_G(GRUPAMENTO_A_ROTULAR.NOH);             ANA0535
                     FUNCAO_G(GRUPAMENTO_A_ROTULAR.NOH) = K;             ANA0536
                END;                                                     ANA0537
                IF J = K                                                 ANA0538
                THEN;                                                    ANA0539
                ELSE HOUVE_MUDANCA_DE_ROTULO = SIM;                      ANA0540
           END;                                                          ANA0541
           CALL OBTER_GRUPAMENTO_A_ROTULAR;                              ANA0542
0     END;    /* FIM DO WHILE PERCORREU_TODOS_GRUPAMENTOS = NAO */       ANA0543
 END;    /* FIM DO WHILE HOUVE_MUDANCA_NO_ROTULO = SIM */                 ANA0544
```

```
1/* COMPLETAR VALOR DAS FUNCOES (5.6)                                    */ANA0545
ODO K = 1 TO NUMERO_DE_SIMBOLOS;                                          ANA0546
      IF F_MARCA_DE_REMOVIDO(K) = REPRESENTANTE                           ANA0547
      THEN                                                                ANA0548
      DO;                                                                 ANA0549
           I = F_PONTEIRO_AO_GRUPAMENTO(K);                               ANA0550
           FUNCAO_DO_REPRESENTANTE = FUNCAO_F(K);                         ANA0551
           DO WHILE(I > 0);                                               ANA0552
                IF TIPO_DE_NOH(I) = F                                     ANA0553
                THEN FUNCAO_F(GRUPAMENTO.NOH(I)) =                        ANA0554
                     FUNCAO_DO_REPRESENTANTE;                             ANA0555
                ELSE FUNCAO_G(GRUPAMENTO.NOH(I)) =                        ANA0556
                     FUNCAO_DO_REPRESENTANTE;                             ANA0557
                I = GRUPAMENTO.PONTEIRO(I);                               ANA0558
           END;                                                           ANA0559
      END;                                                                ANA0560
      ELSE;                                                               ANA0561
 END;                                                                     ANA0562
 DO K = INICIO_G TO NUMERO_DE_SIMBOLOS;                                   ANA0563
      IF G_MARCA_DE_REMOVIDO(K) = REPRESENTANTE                           ANA0564
      THEN                                                                ANA0565
      DO;                                                                 ANA0566
           I = G_PONTEIRO_AO_GRUPAMENTO(K);                               ANA0567
           FUNCAO_DO_REPRESENTANTE = FUNCAO_G(K);                         ANA0568
           DO WHILE(I > 0);                                               ANA0569
                IF TIPO_DE_NOH(I) = F                                     ANA0570
                THEN FUNCAO_F(GRUPAMENTO.NOH(I)) =                        ANA0571
                     FUNCAO_DO_REPRESENTANTE;                             ANA0572
                ELSE FUNCAO_G(GRUPAMENTO.NOH(I)) =                        ANA0573
                     FUNCAO_DO_REPRESENTANTE;                             ANA0574
                I = GRUPAMENTO.PONTEIRO(I);                               ANA0575
           END;                                                           ANA0576
      END;                                                                ANA0577
      ELSE;                                                               ANA0578
 END;                                                                     ANA0579
1/*****************************************************************/     ANA0580
 /*****************  OBTER GRUPAMENTO A ROTULAR (5.3)  ************/     ANA0581
 /*****************************************************************/     ANA0582
OOBTER_GRUPAMENTO_A_ROTULAR:                                              ANA0583
 PROC;                                                                    ANA0584
OIF GRUPAMENTO_A_ROTULAR.TIPO = F                                         ANA0585
 THEN F_INICIO_PROCURA = GRUPAMENTO_A_ROTULAR.NOH + 1;                    ANA0586
 ELSE                                                                     ANA0587
 DO;                                                                      ANA0588
      F_INICIO_PROCURA = NUMERO_DE_SIMBOLOS + 1;                          ANA0589
      G_INICIO_PROCURA = GRUPAMENTO_A_ROTULAR.NOH + 1;                    ANA0590
 END;                                                                     ANA0591
 GRUPAMENTO_A_ROTULAR.NOH = 0;                                            ANA0592
 DO K = F_INICIO_PROCURA TO NUMERO_DE_SIMBOLOS;                           ANA0593
      IF F_MARCA_DE_REMOVIDO(K) = REPRESENTANTE                           ANA0594
      THEN                                                                ANA0595
      DO;                                                                 ANA0596
           GRUPAMENTO_A_ROTULAR.TIPO = F;                                 ANA0597
           PONTEIRO_LIGACOES = F_PONTEIRO_A_LIGACOES(K);                  ANA0598
           GRUPAMENTO_A_ROTULAR.NOH = K;                                  ANA0599
```

```
          K = NUMERO_DE_SIMBOLOS;                                          ANA0600
     END;                                                                  ANA0601
     ELSE;                                                                 ANA0602
 END;                                                                      ANA0603
 IF GRUPAMENTO_A_ROTULAR.NOH = 0                                           ANA0604
 THEN                                                                      ANA0605
 DO K = G_INICIO_PROCURA TO NUMERO_DE_SIMBOLOS;                            ANA0606
     IF G_MARCA_DE_REMOVIDO(K) = REPRESENTANTE                             ANA0607
     THEN                                                                  ANA0608
     DO;                                                                   ANA0609
          GRUPAMENTO_A_ROTULAR.TIPO = G;                                   ANA0610
          PONTEIRO_LIGACOES = G_PONTEIRO_A_LIGACOES(K);                    ANA0611
          GRUPAMENTO_A_ROTULAR.NOH = K;                                    ANA0612
          K = NUMERO_DE_SIMBOLOS;                                          ANA0613
     END;                                                                  ANA0614
     ELSE;                                                                 ANA0615
 END;                                                                      ANA0616
 ELSE;                                                                     ANA0617
 IF GRUPAMENTO_A_ROTULAR.NOH = 0                                           ANA0618
 THEN PERCORREU_TODOS_GRUPAMENTOS = SIM;                                   ANA0619
 ELSE;                                                                     ANA0620
OEND;   /*** FIM DE OBTER_GRUPAMENTO_A_ROTULAR ***/                        ANA0621
OEND;   /*** FIM DE GERAR_VALORES_FUNCOES_LINEARES ***/                    ANA0622
1/*****************************************************************/ANA0623
 /********************* GRAVAR RESULTADOS (6) *********************/ANA0624
 /*****************************************************************/ANA0625
OGRAVAR_RESULTADOS:                                                        ANA0626
 PROC;                                                                     ANA0627
ODCL PAGINA                          BIN FIXED INIT(0),                    ANA0628
     CONTROLES(40)                   BIN FIXED BASED                       ANA0629
                                     (ADDR(CONTROLE)),                     ANA0630
     NUMERO_DE_TERMINAIS             BIN FIXED,                            ANA0631
     NUMERO_DE_PRODUCOES             BIN FIXED,                            ANA0632
     TAMANHO_DOS_ARGUMENTOS          BIN FIXED,                            ANA0633
     PRECEDENCIA                     CHAR(7),                              ANA0634
     (I, J, K)                       BIN FIXED,                            ANA0635
     ARQFUNC                         FILE RECORD OUTPUT;                   ANA0636
-/* DECLARACAO DAS CONSTANTES E FUNCOES                                  */ANA0637
ODCL PRECEDENCIA_FRACA               BIN FIXED INIT(1),                    ANA0638
     ANAGRAMA                        CHAR(22) INIT(                        ANA0639
                          'A  N  A  G  R  A  M  A'),                       ANA0640
     PAG                             CHAR(5) INIT('PAG. '),                ANA0641
     ANALISADOR                      CHAR(43),                             ANA0642
     RELATORIO_DOIS                  CHAR(28) VAR INIT(''),                ANA0643
     ASTERISCO                       CHAR(116) VAR INIT(''),               ANA0644
     CABECALHO                       CHAR(62) VAR INIT(''),                ANA0645
     CARACTERISTICAS                 CHAR(30) INIT(                        ANA0646
                                    'CARACTERISTICAS DA GRAMATICA E'),     ANA0647
     FUNCOES_LINEARES                CHAR(32) INIT(                        ANA0648
                                   ' FUNCOES LINEARES DE PRECEDENCIA'),    ANA0649
     ANALISA                         CHAR(24) INIT(                        ANA0650
                                     'ANALISADOR DE GRAMATICAS'),          ANA0651
     DE_PRECEDENCIA                  CHAR(15) INIT(                        ANA0652
                                     ' DE PRECEDENCIA'),                   ANA0653
     RELATORIO                       CHAR(19) INIT(                        ANA0654
```

```
                                    'RELATORIO 02   -   '),          ANA0655
    DATE                            BUILTIN,                         ANA0656
    ADDR                            BUILTIN,                         ANA0657
    DATA_PLI                        CHAR(6),                         ANA0658
    ANO_MES_DIA(3)                  CHAR(2) BASED                    ANA0659
                                    (ADDR(DATA_PLI)),                ANA0660
    MES                             PIC'99',                         ANA0661
    DATA                            CHAR(9) VAR INIT(''),            ANA0662
    MESES_ANO(12)                   CHAR(3) INIT(                    ANA0663
                          'JAN','FEV','MAR','ABR','MAI','JUN',       ANA0664
                          'JUL','AGO','SET','OUT','NOV','DEZ'),      ANA0665
    TRACO                           CHAR(1) INIT('-'),               ANA0666
    FRACA                           CHAR(7) INIT('FRACA  '),         ANA0667
    SIMPLES                         CHAR(7) INIT('SIMPLES'),         ANA0668
    SIM                             CHAR(1) INIT('S'),               ANA0669
    F                               CHAR(2) INIT('F('),              ANA0670
    G                               CHAR(2) INIT('G(');              ANA0671
-OPEN FILE(SYSPRINT) PAGESIZE (59);                                  ANA0672
1/*****************************************************************/ANA0673
 /*****************  IMPRIMIR CABECALHO (6.1) ************************/ANA0674
 /*****************************************************************/ANA0675
OON ENDPAGE(SYSPRINT)                                                ANA0676
 BEGIN;                                                              ANA0677
      PAGINA = PAGINA + 1;                                           ANA0678
      PUT EDIT(ANAGRAMA, PAG, PAGINA, ANALISADOR, RELATORIO_DOIS)    ANA0679
      (PAGE,SKIP,COL(50),A,COL(108),A,F(3),SKIP(2),COL(41),A,SKIP(2),ANA0680
      COL(47),A);                                                    ANA0681
      PUT EDIT(ASTERISCO,'*',CABECALHO,'*',ASTERISCO)                ANA0682
      (SKIP(3),COL(5),A,COL(5),A,COL(30),A,COL(120),A,COL(5),A);     ANA0683
 END;   /* FIM DO BLOCO BEGIN */                                     ANA0684
-/* FORMATAR CABECALHO (6.2)                                       */ANA0685
ODO K = 1 TO 116;                                                    ANA0686
      ASTERISCO = ASTERISCO CAT '*';                                 ANA0687
 END;                                                                ANA0688
 ANALISADOR = ANALISA CAT DE_PRECEDENCIA;                            ANA0689
 DATA_PLI = DATE;                                                    ANA0690
 MES = ANO_MES_DIA(2);                                               ANA0691
 DATA = ANO_MES_DIA(3) CAT TRACO CAT MESES_ANO(MES)                  ANA0692
        CAT TRACO CAT ANO_MES_DIA(1);                                ANA0693
 RELATORIO_DOIS = RELATORIO CAT DATA;                                ANA0694
 CABECALHO = CARACTERISTICAS CAT FUNCOES_LINEARES;                   ANA0695
 SIGNAL ENDPAGE(SYSPRINT);                                           ANA0696
1/* IMPRIMIR CARACTERISTICAS DA GRAMATICA (6.3)                    */ANA0697
ONUMERO_DE_PRODUCOES = CONTROLES(4);                                 ANA0698
 TAMANHO_DOS_ARGUMENTOS = CONTROLES(5);                              ANA0699
 NUMERO_DE_TERMINAIS = NUMERO_DE_SIMBOLOS - NAOTERMINAIS;            ANA0700
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                     ANA0701
 THEN PRECEDENCIA = FRACA;                                           ANA0702
 ELSE PRECEDENCIA = SIMPLES;                                         ANA0703
 PUT EDIT('CARACTERISTICAS DA GRAMATICA',                            ANA0704
          '--------------------------',                             ANA0705
          '- NUMERO DE PRODUCOES: ', NUMERO_DE_PRODUCOES,            ANA0706
          '- NUMERO DE TERMINAIS: ', NUMERO_DE_TERMINAIS,            ANA0707
          '- NUMERO DE NAO TERMINAIS: ', NAOTERMINAIS,               ANA0708
          '- TAMANHO ACUMULADO DOS ARGUMENTOS: ',TAMANHO_DOS_ARGUMENTOS,ANA0709
```

```
          '= TIPO DE PRECEDENCIA: ',PRECEDENCIA)                    ANA0710
 (SKIP(5), COL(26), A, SKIP, COL(26), A,                            ANA0711
 4(SKIP(2),COL(31),A,F(4)),                                         ANA0712
 SKIP(2),COL(31),A,A);                                              ANA0713
-/* IMPRIMIR FUNCOES (6.4)                                         */ANA0714
OPUT EDIT('VALORES DAS FUNCOES LINEARES',                           ANA0715
          '--------------------------')                            ANA0716
 (SKIP(5),COL(26),A,SKIP,COL(26),A);                                ANA0717
 IF INDICADOR_DE_CICLOS = SIM                                       ANA0718
 THEN PUT EDIT( '*** NAO EXISTEM FUNCOES LINEARES PARA A ',         ANA0719
                'MATRIZ DE PRECEDENCIA DESTA GRAMATICA ***')        ANA0720
      (SKIP(2),COL(31),A,A);                                        ANA0721
 ELSE                                                               ANA0722
 DO;                                                                ANA0723
      PUT EDIT('FUNCAO F :')(SKIP(2),COL(31),A);                    ANA0724
      CALL FORMATAR_VALORES(FUNCAO_F, F);                           ANA0725
      PUT EDIT('FUNCAO G :')(SKIP(2),COL(31),A);                    ANA0726
      CALL FORMATAR_VALORES(FUNCAO_G, G);                           ANA0727
 END;                                                               ANA0728
1/* GRAVAR ARQUIVO DE FUNCOES (6.5)                                */ANA0729
OIF INDICADOR_DE_CICLOS = SIM                                       ANA0730
 THEN CONTROLES(8) = 0;                                             ANA0731
 ELSE                                                               ANA0732
 DO;                                                                ANA0733
      CONTROLES(8) = 1;                                             ANA0734
      WRITE FILE (ARQFUNC) FROM (CONTROLE);                         ANA0735
      J = 0;                                                        ANA0736
      DO K = 1 TO NUMERO_DE_SIMBOLOS;                               ANA0737
           J = J + 1;                                               ANA0738
           IF J > 40                                                ANA0739
           THEN                                                     ANA0740
           DO;                                                      ANA0741
                WRITE FILE(ARQFUNC) FROM (CONTROLE);                ANA0742
                J = 1;                                              ANA0743
           END;                                                     ANA0744
           ELSE;                                                    ANA0745
           CONTROLES(J) = FUNCAO_F(K);                              ANA0746
      END;                                                          ANA0747
      IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA               ANA0748
      THEN I = NAOTERMINAIS + 1;                                    ANA0749
      ELSE I = 1;                                                   ANA0750
      DO K = I TO NUMERO_DE_SIMBOLOS;                               ANA0751
           J = J + 1;                                               ANA0752
           IF J > 40                                                ANA0753
           THEN                                                     ANA0754
           DO;                                                      ANA0755
                WRITE FILE (ARQFUNC) FROM (CONTROLE);               ANA0756
                J = 1;                                              ANA0757
           END;                                                     ANA0758
           ELSE;                                                    ANA0759
           CONTROLES(J) = FUNCAO_G(K);                              ANA0760
      END;                                                          ANA0761
      WRITE FILE (ARQFUNC) FROM (CONTROLE);                         ANA0762
END;                                                                ANA0763
```

```
1/*****************************************************************************/ANA0764
 /*********************** FORMATAR VALORES (6.6) ***************************/ANA0765
 /*****************************************************************************/ANA0766
OFORMATAR_VALORES:                                                             ANA0767
 PROC (FUNCAO, TIPO_DA_FUNCAO);                                                ANA0768
ODCL FUNCAO(*)                          BIN FIXED,                             ANA0769
     TIPO_DA_FUNCAO                     CHAR(2);                               ANA0770
OI, J, K = 1;                                                                  ANA0771
 IF TIPO_DA_FUNCAO = G                                                         ANA0772
 THEN IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                          ANA0773
      THEN J = NAOTERMINAIS + 1;                                               ANA0774
      ELSE;                                                                    ANA0775
 ELSE;                                                                         ANA0776
 PUT EDIT(' ')(SKIP,COL(30),A);                                                ANA0777
 DO K = J TO NUMERO_DE_SIMBOLOS;                                               ANA0778
      IF I = 1                                                                 ANA0779
      THEN;                                                                    ANA0780
      ELSE PUT EDIT(',  ')(A);                                                 ANA0781
      PUT EDIT(TIPO_DA_FUNCAO, K, ')=',FUNCAO(K))(A,F(3),A,F(3));              ANA0782
      I = I + 1;                                                               ANA0783
      IF I > 7                                                                 ANA0784
      THEN                                                                     ANA0785
      DO;                                                                      ANA0786
           PUT EDIT(' ')(SKIP,COL(30),A);                                      ANA0787
           I = 1;                                                              ANA0788
      END;                                                                     ANA0789
      ELSE;                                                                    ANA0790
 END;                                                                          ANA0791
OEND;   /* FIM DE IMPRIMIR_VALORES_DAS_FUNCOES */                              ANA0792
OEND;   /*** FIM DE GRAVAR_RESULTADOS ***/                                     ANA0793
-END;   /*** FIM DE ANAG201 ***/                                              ANA0794
```

Peço licença para terminar
soletrando a canção de rebeldia
que existe nos fonemas da alegria:
canção de amor geral que eu vi crescer
nos olhos do homem que aprendeu a ler.

"Canção para os fonemas de alegria"

THIAGO DE MELO

CAPÍTULO VI

COMPACTAÇÃO DA MATRIZ DE PRECEDÊNCIA

Quando as relações de precedência de uma gramática G não podem ser representadas por funções lineares ou quando estas não são consideradas eficientes no reconhecedor, pode-se usar a matriz de precedência. Neste caso, o espaço de armazenamento pode tornar-se crítico, dado que o número de entradas pode chegar a n^2, onde n é o número de símbolos de G. A matriz de precedência tem uma particularidade: suas entradas podem conter, no máximo, quatro tipos de valores (MENOR, IGUAL, MAIOR ou "erro") e por causa disto, as repetições de valores ou de grupos de valores são muito frequentes. Podem ocorrer vários grupos de linhas ou colunas iguais, ou então, dentro de uma mesma linha ou coluna, longos trechos preenchidos com um mesmo valor. Nasce daí a idéia de compactar a matriz de precedência, encontrando uma forma de armazenamento que seja econômica em termos de espaço e eficiente em termos de recuperação. Os procedimentos a seguir descritos fornecem três formas de compactar a matriz de precedência, de modo que o projetista de um reconhecedor possa escolher aquela mais adequada a seu caso.

VI.1 - Entrada para a Compactação

Uma matriz de precedência M, com n linhas e m colunas.

VI.2 - Saídas da Compactação

A matriz de precedência M é compactada de três maneiras, obtendo-se as seguintes estruturas de compactação.

(1) matriz reduzida;

(2) matriz original compactada;

(3) matriz reduzida compactada.

Cada uma destas estruturas é descrita a seguir.

VI.2.1 - Matriz Reduzida

Esta forma de compactação reduz as linhas e colunas iguais da matriz M, de n linhas e m colunas, gerando uma matriz reduzida R, de r linhas e s colunas. Desta maneira, uma linha de R representa uma ou mais linhas iguais de M, valendo o mesmo para as colunas. A matriz M é compactada então na estrutura Q = (R, A, B), onde:

(1) R é a matriz reduzida de M;

(2) $A = (a_1, a_2, \ldots a_n)$ é o vetor redutor de linhas, tal que a_i indica a linha de R que representa a linha i de M;

(3) $B = (b_1, b_2, \ldots b_m)$ é o vetor redutor de colunas, tal que b_j indica a coluna de R que representa a coluna j de M.

Um elemento M_{ij} é recuperado na estrutura Q da seguinte forma:

$$M_{ij} = R_{a_i\ b_j}$$

Por exemplo, seja a matriz de precedência M:

$$\begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Nesta matriz 1, 2 e 3 representam respectivamente MENOR, IGUAL e MAIOR. A matriz reduzida R é

$$\begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & 2 \\ 0 & 3 & 3 \end{vmatrix}$$

e os vetores redutores de linha e coluna, A e B, são os seguintes:

$$A = (1, 2, 3, 1)$$

$$B = (1, 2, 1, 3).$$

Por exemplo, $M_{43} = R_{11} = 0$.

As gramáticas analisadas pelo ANAGRAMA podem ter, no máximo, 255 símbolos (como foi visto no capítulo III). Logo 256 será a maior dimensão possível de M. Como consequência A ou B, terão no máximo 256 elementos e cada valor a_i ou b_j pode ser descrito em 1 byte. Cada elemento de M e R pode ser descrito em 2 bits. O fator de compactação obtido por R é dado, então, por

$$\frac{rs + 4\ (m + n)}{mn}$$

VI.2.2 - Matriz Original Compactada

Uma matriz pode ser compactada criando-se uma representação abreviada para os grupos de elementos vizinhos, iguais e dentro de uma mesma linha. Esta representação é um

par (y, M_{ij}), onde M_{ij} é o elemento da matriz que se repete e y é a ordem da coluna do último elemento do grupo. Por exemplo, seja a matriz M dada por

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 1 & 1 & 0 \end{vmatrix}$$

O grupo de três 0's da primeira linha pode ser representado pelo par (4, 0). Uma matriz M de n linhas e m colunas pode ser representada pela dupla C = (P, L), onde

(1) $P = (p_1, p_2, \ldots p_w)$ é a sequência de pares obtida na compactação de todas as linhas de M, como foi descrito atrás, e w é o número de pares obtidos;

(2) $L = (l_1, l_2 \ldots l_n)$ é a sequência de ponteiros aos primeiros pares de cada linha, ou seja, l_i é a ordem do primeiro par da linha i.

A matriz M atrás referida pode ser representad pela dupla C = (P, L), onde

P = ((1, 1), (4, 0), (3, 1), (4, 0), (2, 0), (4, 2), (1, 0), (3, 1), (4, 0)),

L = (1, 3, 5, 7).

Um elemento M_{ij} pode ser recuperado da seguinte maneira:

1 - Obter $a = l_i$ em L;

2 - Obter $P_k = (y, M_{iy})$, para $K \geq a$, tal que $j \leq y$;

M_{iy} é o elemento procurado.

No dimensionamento de L é admitido que o número de pares nunca é maior que 32767 (se for, certamente esta forma de compactação não é a mais econômica). Logo, cada elemento l_i em L ocupará 2 bytes. O valor y de cada par ocupa 1 byte e os valores M_{ij} ocupam 2 bits. O fator de compactação resultante de C é dado por

$$\frac{5w + 8n}{mn}$$

Este fator pode ser melhorado se $w \leq 256$, quando então cada l_i pode ocupar 1 byte, ao invés de 2. O numerador da fórmula acima torna-se então

$$5w + 4n$$

VI.2.3 - Matriz Reduzida Compactada

O mesmo processo de compactação aplicado a matriz original é repetido à matriz reduzida R, obtida como descrito em VI.2.1. R passa a ser representada por C = (L, P). No entanto, L se refere às r linhas de R, as quais são relacionadas às n linhas da matriz original M pelo vetor redutor de linha $A = (a_1, a_2, \ldots a_n)$. É criado então o vetor H assim definido:

$H = (h_1, h_2, \ldots h_n)$ tal que

$h_i = l_k$, $k = a_i$, para $1 \leq i \leq n$.

Este vetor H indexa os pares gerados pela compactação da reduzida, fazendo a necessária correspondência entre as linhas da reduzida e da original. A matriz de precedência M experimenta então dois processos sucessivos de compactação e pode ser representada pela tripla T = (H, B, P), onde

(1) H é o vetor definido atrás;

(2) B é o redutor de colunas da matriz reduzida;

(3) P é a sequência de pares obtidos da compactação da matriz reduzida.

Tomando a matriz M do item VI.2.1 e sua reduzida R tem-se que:

H = (1, 3, 6, 1);

B = (1, 2, 1, 3);

P = ((2, 0), (3, 1), (1, 1), (2, 0), (3, 2), (1, 0), (3, 3)).

No vetor H, por exemplo, $h_4 = 1$ significa que o primeiro par da linha 4 da matriz original M, é dado pelo par 1 em P. Vale a pena repetir que P é gerado pela compactação da reduzida R. O vetor B é o redutor de colunas obtido como mostrado em VI.2.1. Um elemento M_{ij} é recuperada da seguinte maneira:

1 - Obter $a = h_i$ em H;

2 - Obter $c = b_j$ em B;

3 - Obter $p_k = (y, R_{uy})$, para $K \geq a$, tal que $c \leq y$;

R_{uy} é o elemento procurado.

O fator de compactação obtido por T é dado por

$$\frac{5w + 8n + 4m}{mn}$$

Como foi explicado no item anterior, o numerador pode tornar-se $5w + 4n + 4m$, se $w \leq 256$. Aqui, w é o número de pares obtidos da compactação de R.

VI.3 - Descrição do Método

São usadas básicamente três algoritmos e o método é descrito na seguinte sequência de passos:

Passo 1 : Construir o vetor redutor de linhas

$A = (a_1, a_2, \ldots a_n)$ usando o Algoritmo R1.

Passo 2: Construir o vetor redutor de colunas

$B = (b_1, b_2, \ldots b_m)$ usando o Algoritmo R_2.

Passo 3: Compactar a matriz original M e a reduzida R utilizando o Algoritmo C e gerando para cada uma delas uma dupla (L, P). Para isto, cada linha de M é lida e, a partir de A e B, são construidas as linhas da matriz reduzida R.

Passo 4: A partir de A e da dupla (L, P) obtida da compactação de R, construir

$H = (h_i, h_2, \ldots h_n)$ onde $h_i = i_k$, $k = a_i$

Passo 5: Calcular os fatores de compactação para a matriz reduzida, a matriz original compactada e a matriz reduzida compactada.

VI.3.1 - Algoritmo R1 - Redução de Linhas

Passo 1: Criar uma lista L_1 contendo n elementos, cada um representando uma linha M.

Passo 2: Obter a lista L_i. Seja l a linha representada em um ele

mento de L_i escolhido arbitrariamente. Excluir de L_i os elementos j, tais que a linha j seja diferente da linha 1 e criar uma nova lista com estes elementos excluidos. Repetir este passo para todas as listas L_i.

Passo 3: Criar um vetor $A = (a_1, a_2 \ldots a_n)$, n o número de linhas de M. Para todo a_j fazer $a_j = i$, onde j é um elemento de L_i.

VI.3.2 - Algoritmo R2 - Redução de Colunas

Passo 1: Criar uma lista L_i contendo m elementos, cada um representando uma coluna de M.

Passo 2: Se o número de listas criadas é menor que o número de linhas de M então obter a linha k, senão executar o passo 4.

Passo 3: Obter a lista L_i. Seja c a coluna representada em um elemento de L_i escolhido arbitrariamente. Excluir de L_i os elementos j, tais que, na linha k, a coluna j seja diferente da coluna c e criar uma nova lista com estes elementos excluidos. Repetir este passo para todas as listas L_i e em seguida executar o passo 2 para o próximo k.

Passo 4: Criar um vetor $B = (b_1, b_2 \ldots b_m)$, m o número de colunas de M. Para todo b_j fazer $b_j = i$, onde j é um elemento de L_i.

VI.3.3 - Algoritmo C-(Compactação da Matriz)

Passo 1: Obter a linha i da matriz X a ser compactada.

Fazer $q = X_{il}$

Passo 2: Obter a coluna j da linha i.

Se $X_{ij} \neq q$ então

criar o par $P_w = (j - 1, q)$ e

fazer $q = M_{ij}$.

Repetir este passo para todas as colunas j.

Passo 3: Criar o par $p_w = (c, M_{ic})$, onde c é o número de colunas de X. Se a linha i não é a última então executar o passo 1 para o próximo i.

VI.4 - Implementação do Módulo

A compactação da matriz de precedência é feita no Módulo 03 do ANAGRAMA. Este módulo é implementado como um programa de nome ANAG301. Os recursos de memória são fixos, independendo pois da gramática sendo analisada. As figuras deste capítulo documentam o programa ANAG301, do qual é fornecida uma listagem em imagem de cartões.

ANAGRAMA - MODULO 03 - COMPACTAR A MATRIZ DE PRECEDENCIA

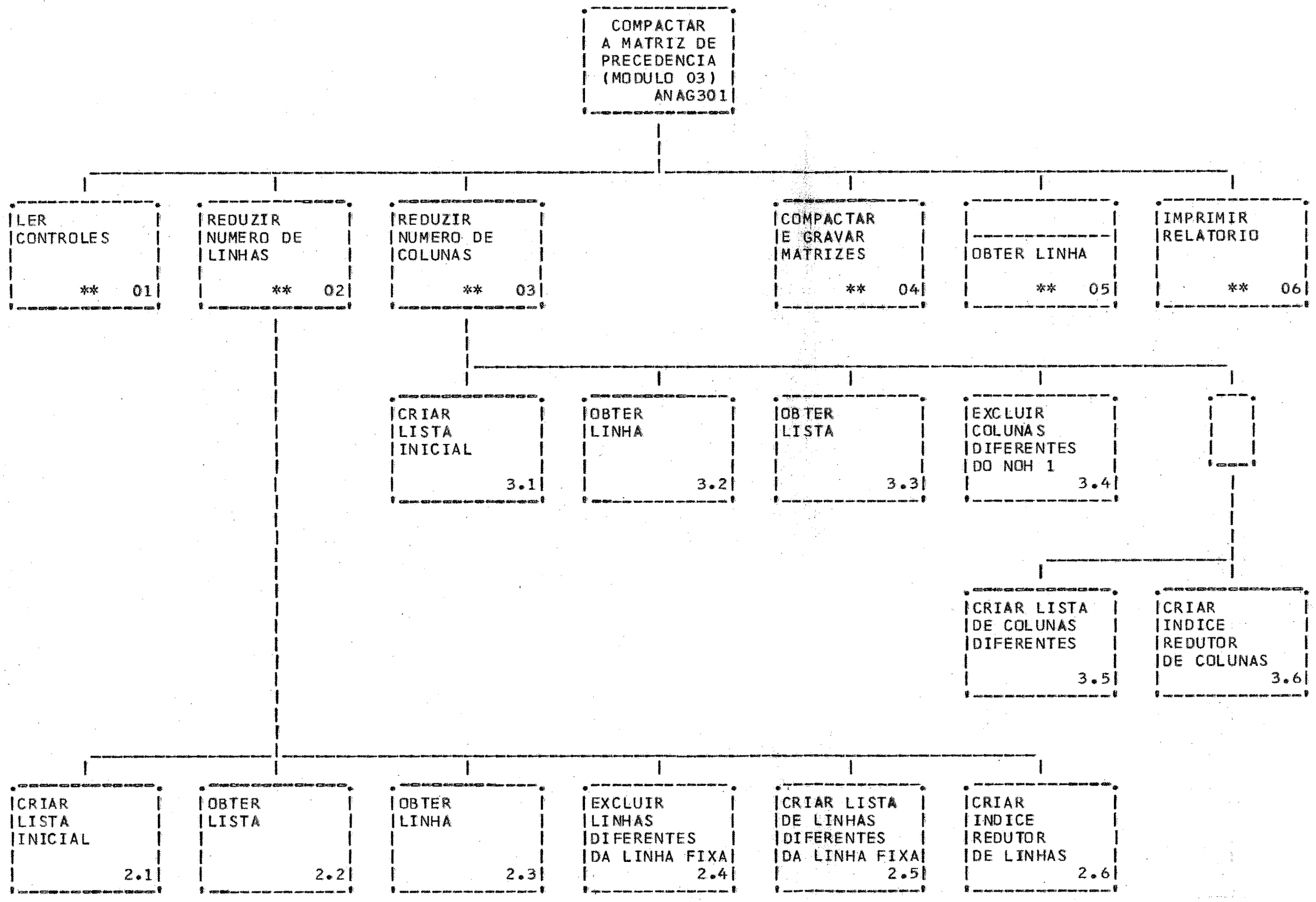
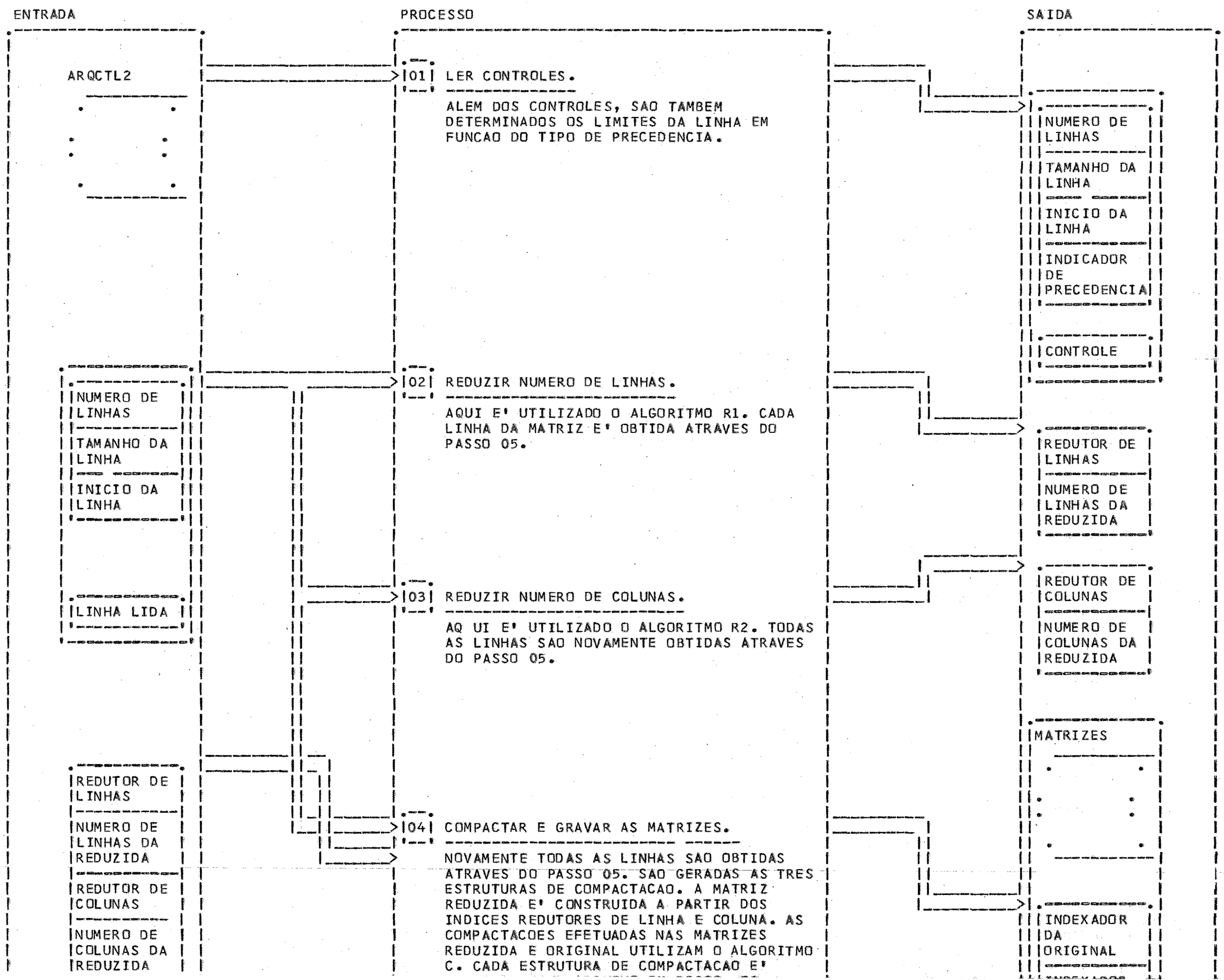


FIGURA VI.1

ANAGRAMA - MODULO 03 - COMPACTAR A MATRIZ DE PRECEDENCIA



GRAVADA EM UM ARQUIVO EM DISCO. E' TAMBEM GRAVADA A MATRIZ NAO COMPACTADA.

ARQMAT

INDICADOR DE PRECEDENCIA

TAMANHO DA LINHA

INICIO DA LINHA

NUMERO DA LINHA

05 OBTER LINHA.

CADA LINHA DA MATRIZ ESTA' EM UM REGISTRO DE 256 CARACTERES. CADA LINHA E' LIDA COM ACESSO DIRETO. A CHAVE DE ACESSO E' A ORDEM DA LINHA. SE A PRECEDENCIA E' FRACA 'IGUAL' E 'IGUAL/MENOR' SAO CONVERTIDOS PARA 'MENOR' E A LINHA GERADA CORRESPONDE APENAS AOS TERMINAIS.

LINHA LIDA

INDEXADOR DA REDUZIDA

NUMERO DE PARES DA ORIGINAL COMPACTADA

NUMERO DE PARES DA REDUZIDA COMPACTADA

CONTROLE

NUMERO DE LINHAS DA REDUZIDA

NUMERO DE COLUNAS DA REDUZIDA

NUMERO DE PARES DA ORIGINAL COMPACTADA

NUMERO DE PARES DA REDUZIDA COMPACTADA

06 IMPRIMIR RELATORIO.

A PARTIR DOS INDICADORES GERADOS NOS PASSOS 2, 3 E 4 SAO CALCULADAS E IMPRESSAS AS TAXAS DE COMPACTACAO. AS CARACTERISTICAS DA GRAMATICA SAO TAMBEM IMPRESSAS.

RELATORIO 3

FIGURA VI.2

```
        ****************************************
        *  LISTAGEM FONTE DO PROGRAMA ANAG301  *
        ****************************************


 /* COMPACTAR A MATRIZ DE PRECEDENCIA - VERSAO DE 10/07/79            */ANA0001
=ANAG301:                                                               ANA0002
 PROC OPTIONS (MAIN);                                                   ANA0003
ODCL REDUTOR_DE_LINHAS(256)             BIN FIXED,                      ANA0004
     NUMERO_LINHAS_DA_REDUZIDA          BIN FIXED,                      ANA0005
     REDUTOR_DE_COLUNAS(256)            BIN FIXED,                      ANA0006
     NUMERO_COLUNAS_DA_REDUZIDA         BIN FIXED,                      ANA0007
     INDICADOR_DE_PRECEDENCIA           BIN FIXED,                      ANA0008
     NUMERO_PARES_ORIGINAL_COMP         BIN FIXED,                      ANA0009
     NUMERO_PARES_REDUZIDA_COMP         BIN FIXED,                      ANA0010
     NUMERO_DE_LINHAS                   BIN FIXED,                      ANA0011
     TAMANHO_DA_LINHA                   BIN FIXED,                      ANA0012
     INICIO_DA_LINHA                    BIN FIXED;                      ANA0013
ODCL LINHA_LIDA                         CHAR(256),                      ANA0014
     COLUNA(256)                        CHAR(1) BASED(ADDR(LINHA_LIDA)), ANA0015
     LINHA_ORIGINAL(256)                CHAR(1) BASED(ADDR(LINHA_LIDA)), ANA0016
     (I, J, K)                          BIN FIXED,                      ANA0017
     CONTROLE                           CHAR(80),                       ANA0018
     ARQMAT                             FILE RECORD DIRECT INPUT KEYED  ANA0019
                                        ENV (REGIONAL(1));              ANA0020
ODCL ADDR                               BUILTIN;                        ANA0021
1/*****************************************************************/ANA0022
 /********************* LER CONTROLES (1) *************************/ANA0023
 /*****************************************************************/ANA0024
OBEGIN;                                                                 ANA0025
ODCL CONTROLES(40)                      BIN FIXED BASED(ADDR(CONTROLE)), ANA0026
     NAOTERMINAIS                       BIN FIXED,                      ANA0027
     ARQCTL2                            FILE RECORD INPUT;              ANA0028
ODCL PRECEDENCIA_FRACA                  BIN FIXED INIT(1);              ANA0029
 READ FILE (ARQCTL2) INTO (CONTROLE);                                   ANA0030
 NUMERO_DE_LINHAS, TAMANHO_DA_LINHA = CONTROLES(3);                     ANA0031
 INDICADOR_DE_PRECEDENCIA = CONTROLES(7);                               ANA0032
 NAOTERMINAIS = CONTROLES(2);                                           ANA0033
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0034
 THEN INICIO_DA_LINHA = NAOTERMINAIS + 1;                               ANA0035
 ELSE INICIO_DA_LINHA = 1;                                              ANA0036
OEND;    /* FIM DO BLOCO BEGIN */                                       ANA0037
1/*****************************************************************/ANA0038
 /******************** REDUZIR NUMERO DE LINHAS (2) *******************/ANA0039
 /*****************************************************************/ANA0040
OBEGIN;                                                                 ANA0041
ODCL CABECA_DE_LISTA(256)               BIN FIXED,                      ANA0042
     NOH(256)                           BIN FIXED,                      ANA0043
     PONTEIRO(256)                      BIN FIXED,                      ANA0044
     NUMERO_DE_LISTAS                   BIN FIXED INIT(1);              ANA0045
ODCL LINHA_FIXA                         CHAR(256),                      ANA0046
     EXCLUIDOS(256)                     BIN FIXED,                      ANA0047
     NUMERO_DE_EXCLUIDOS                BIN FIXED,                      ANA0048
     ANTERIOR                           BIN FIXED;                      ANA0049
-/* CRIAR LISTA INICIAL (2.1)                                          */ANA0050
OCABECA_DE_LISTA(1) = 1;                                                ANA0051
 DO K = 1 TO NUMERO_DE_LINHAS;                                          ANA0052
      NOH(K) = K;                                                       ANA0053
      PONTEIRO(K) = K + 1;                                              ANA0054
 END;                                                                   ANA0055
```

```
 PONTEIRO(NUMERO_DE_LINHAS) = 0;                                        ANA0056
 I = 0;                                                                 ANA0057
1DO WHILE(I < NUMERO_DE_LISTAS);                                        ANA0058
0/* OBTER LISTA (2.2)                                                  */ANA0059
0     I = I + 1;                                                        ANA0060
      J = CABECA_DE_LISTA(I);                                           ANA0061
-/* OBTER LINHA FIXA (2.3)                                             */ANA0062
0     K = NOH(J);                                                       ANA0063
      CALL OBTER_LINHA(K);                                              ANA0064
      LINHA_FIXA = LINHA_LIDA;                                          ANA0065
-/* EXLUIR DA LISTA LINHAS DIFERENTES DA LINHA FIXA (2.4)              */ANA0066
0     NUMERO_DE_EXCLUIDOS = 0;                                          ANA0067
      ANTERIOR = J;                                                     ANA0068
      J = PONTEIRO(J);                                                  ANA0069
      DO WHILE(J > 0);                                                  ANA0070
           K = NOH(J);                                                  ANA0071
           CALL OBTER_LINHA(K);                                         ANA0072
           IF LINHA_LIDA ¬= LINHA_FIXA                                  ANA0073
           THEN                                                         ANA0074
           DO;                                                          ANA0075
                PONTEIRO(ANTERIOR) = PONTEIRO(J);                       ANA0076
                NUMERO_DE_EXCLUIDOS = NUMERO_DE_EXCLUIDOS + 1;          ANA0077
                EXCLUIDOS(NUMERO_DE_EXCLUIDOS) = J;                     ANA0078
           END;                                                         ANA0079
           ELSE ANTERIOR = J;                                           ANA0080
           J = PONTEIRO(J);                                             ANA0081
      END;                                                              ANA0082
-/* CRIAR LISTA DE LINHAS DIFERENTES DA LINHA FIXA (2.5)               */ANA0083
0     IF NUMERO_DE_EXCLUIDOS > 0                                        ANA0084
      THEN                                                              ANA0085
      DO;                                                               ANA0086
           NUMERO_DE_LISTAS = NUMERO_DE_LISTAS + 1;                     ANA0087
           CABECA_DE_LISTA(NUMERO_DE_LISTAS) = EXCLUIDOS(1);            ANA0088
           DO K = 1 TO NUMERO_DE_EXCLUIDOS;                             ANA0089
                PONTEIRO(EXCLUIDOS(K)) = EXCLUIDOS(K + 1);              ANA0090
           END;                                                         ANA0091
           PONTEIRO(EXCLUIDOS(NUMERO_DE_EXCLUIDOS)) = 0;                ANA0092
      END;                                                              ANA0093
      ELSE;                                                             ANA0094
 END;   /* FIM DO WHILE ... */                                         ANA0095
1/* CRIAR INDICE REDUTOR DE LINHAS (2.6)                               */ANA0096
0DO K = 1 TO NUMERO_DE_LISTAS;                                          ANA0097
      J = CABECA_DE_LISTA(K);                                           ANA0098
      DO WHILE(J > 0);                                                  ANA0099
           REDUTOR_DE_LINHAS(NOH(J)) = K;                               ANA0100
           J = PONTEIRO(J);                                             ANA0101
      END;                                                              ANA0102
 END;                                                                   ANA0103
 NUMERO_LINHAS_DA_REDUZIDA = NUMERO_DE_LISTAS;                          ANA0104
0END;   /* FIM DO BLOCO BEGIN */                                        ANA0105
1/*****************************************************************/ANA0106
 /****************** REDUZIR NUMERO DE COLUNAS (3) ********************/ANA0107
 /*****************************************************************/ANA0108
0BEGIN;                                                                 ANA0109
0DCL CABECA_DE_LISTA(256)          BIN FIXED,                           ANA0110
     NOH(256)                      BIN FIXED,                           ANA0111
     PONTEIRO(256)                 BIN FIXED,                           ANA0112
     NUMERO_DE_LISTAS              BIN FIXED INIT(1);                   ANA0113
0DCL EXCLUIDOS(256)                BIN FIXED,                           ANA0114
     NUMERO_DE_EXCLUIDOS           BIN FIXED,                           ANA0115
```

```
     PRIMEIRO_NOH                        BIN FIXED,                          ANA0116
     ANTERIOR                            BIN FIXED;                          ANA0117
-/* CRIAR LISTA INICIAL (3.1)                                               */ANA0118
0CABECA_DE_LISTA(1) = INICIO_DA_LINHA;                                       ANA0119
 DO K = INICIO_DA_LINHA TO TAMANHO_DA_LINHA;                                 ANA0120
      NOH(K) = K;                                                            ANA0121
      PONTEIRO(K) = K + 1;                                                   ANA0122
 END;                                                                        ANA0123
 PONTEIRO(TAMANHO_DA_LINHA) = 0;                                             ANA0124
1DO K = 1 TO NUMERO_DE_LINHAS;                                               ANA0125
0/* OBTER LINHA (3.2)                                                       */ANA0126
0     I = 0;                                                                 ANA0127
      IF NUMERO_DE_LISTAS < NUMERO_DE_LINHAS                                 ANA0128
      THEN CALL OBTER_LINHA( K );                                            ANA0129
      ELSE I, K = NUMERO_DE_LINHAS;                                          ANA0130
      DO WHILE (I < NUMERO_DE_LISTAS);                                       ANA0131
-/* OBTER LISTA (3.3)                                                       */ ANA0132
           I = I + 1;                                                        ANA0133
           J = CABECA_DE_LISTA(I);                                           ANA0134
-/* EXCLUIR COLUNAS DIFERENTES DA COLUNA DO PRIMEIRO NOH (3.4)              */ANA0135
0          NUMERO_DE_EXCLUIDOS = 0;                                          ANA0136
           PRIMEIRO_NOH = NOH(J);                                            ANA0137
           DO WHILE(J > 0);                                                  ANA0138
                IF COLUNA(PRIMEIRO_NOH) ¬= COLUNA(NOH(J))                    ANA0139
                THEN                                                         ANA0140
                DO;                                                          ANA0141
                     PONTEIRO(ANTERIOR) = PONTEIRO(J);                       ANA0142
                     NUMERO_DE_EXCLUIDOS = NUMERO_DE_EXCLUIDOS + 1;          ANA0143
                     EXCLUIDOS(NUMERO_DE_EXCLUIDOS) = J;                     ANA0144
                END;                                                         ANA0145
                ELSE ANTERIOR = J;                                           ANA0146
                J = PONTEIRO(J);                                             ANA0147
           END;                                                              ANA0148
-/* CRIAR LISTA COM COLUNAS DIFERENTES DA COLUNA DO PRIMEIRO NOH (3.5)*/     ANA0149
0          IF NUMERO_DE_EXCLUIDOS > 0                                        ANA0150
           THEN                                                              ANA0151
           DO;                                                               ANA0152
                NUMERO_DE_LISTAS = NUMERO_DE_LISTAS + 1;                     ANA0153
                CABECA_DE_LISTA(NUMERO_DE_LISTAS) = EXCLUIDOS(1);            ANA0154
                DO J = 1 TO NUMERO_DE_EXCLUIDOS;                             ANA0155
                     PONTEIRO(EXCLUIDOS(J)) = EXCLUIDOS(J + 1);              ANA0156
                END;                                                         ANA0157
                PONTEIRO(EXCLUIDOS(NUMERO_DE_EXCLUIDOS)) = 0;                ANA0158
           END;                                                              ANA0159
           ELSE;                                                             ANA0160
      END;   /* FIM DO WHILE ... */                                         ANA0161
 END;   /* FIM DO K = 1 ... */                                               ANA0162
1/* CRIAR INDICE REDUTOR DE COLUNAS (3.6)                                  */ANA0163
0DO K = 1 TO NUMERO_DE_LISTAS;                                               ANA0164
      J = CABECA_DE_LISTA(K);                                                ANA0165
      DO WHILE(J > 0);                                                       ANA0166
           REDUTOR_DE_COLUNAS(NOH(J)) = K;                                   ANA0167
           J = PONTEIRO(J);                                                  ANA0168
      END;                                                                   ANA0169
 END;                                                                        ANA0170
 NUMERO_COLUNAS_DA_REDUZIDA = NUMERO_DE_LISTAS;                              ANA0171
0END;   /* FIM DO BLOCO BEGIN */                                             ANA0172
1/*************************************************************************/ANA0173
 /*************** COMPACTAR E GRAVAR MATRIZES (4) ***********************/ANA0174
 /*************************************************************************/ANA0175
```

```
OBEGIN ;                                                                ANA0176
ODCL  INDEX_ORIGINAL(257)          BIN FIXED INIT(0),                   ANA0177
      INDEX_REDUZIDA(257)          BIN FIXED,                           ANA0178
      INDEX_RED(257)               BIN FIXED INIT(0);                   ANA0179
ODCL  LINHA_REDUZIDA(256)          CHAR(1),                             ANA0180
      N_ORIGINAL                   BIN FIXED INIT(0),                   ANA0181
      N_REDUZIDA                   BIN FIXED INIT(0),                   ANA0182
      ULTIMA_LINHA_REDUZIDA        BIN FIXED INIT(0),                   ANA0183
      ORIGINAL                     CHAR(1) INIT('O'),                   ANA0184
      REDUZIDA                     CHAR(1) INIT('R'),                   ANA0185
      COMPACTA_REDUZIDA            CHAR(1),                             ANA0186
      N                            BIN FIXED INIT(0);                   ANA0187
ODCL  REG_ORIGINAL                 CHAR(80),                            ANA0188
      1 PAR_ORIGINAL               BASED(ADDR(REG_ORIGINAL)),           ANA0189
        2 A(64)                    CHAR(1),                             ANA0190
        2 B(64)                    BIT(2),                              ANA0191
      VETOR_BINFIXED(40)           BIN FIXED BASED                      ANA0192
                                   (ADDR(REG_ORIGINAL)),                ANA0193
      VETOR_CHAR(80)               CHAR(1) BASED(ADDR(REG_ORIGINAL)),ANA0194
      REG_REDUZIDA                 CHAR(80),                            ANA0195
      1 PAR_REDUZIDA               BASED(ADDR(REG_REDUZIDA)),           ANA0196
        2 A(64)                    CHAR(1),                             ANA0197
        2 B(64)                    BIT(2),                              ANA0198
      REG_MAT_ORIGINAL             CHAR(80),                            ANA0199
      REG_MAT_REDUZIDA             CHAR(80),                            ANA0200
      ARQPREC                      FILE RECORD OUTPUT,                  ANA0201
      ARQRED                       FILE RECORD OUTPUT,                  ANA0202
      ARQCP1                       FILE RECORD OUTPUT,                  ANA0203
      ARQCP2                       FILE RECORD OUTPUT;                  ANA0204
ODCL  UM                           BIN FIXED INIT(1),                   ANA0205
      SIM                          CHAR(1) INIT('S'),                   ANA0206
      NAO                          CHAR(1) INIT('N');                   ANA0207
1/* GRAVAR CONTROLES DAS MATRIZES (4.1)                                 */ANA0208
0CALL GRAVAR_REGISTRO(SIM);                                              ANA0209
 DO K = 1 TO NUMERO_DE_LINHAS;                                           ANA0210
-/* OBTER LINHA ORIGINAL (4.2)                                          */ANA0211
0     CALL OBTER_LINHA(K);                                               ANA0212
-/* REDUZIR A LINHA (4.3)                                               */ANA0213
0     IF REDUTOR_DE_LINHAS(K) > ULTIMA_LINHA_REDUZIDA                    ANA0214
      THEN                                                               ANA0215
      DO;                                                                ANA0216
           DO J = INICIO_DA_LINHA TO TAMANHO_DA_LINHA;                   ANA0217
                LINHA_REDUZIDA(REDUTOR_DE_COLUNAS(J))=LINHA_ORIGINAL(J);ANA0218
           END;                                                          ANA0219
           ULTIMA_LINHA_REDUZIDA = ULTIMA_LINHA_REDUZIDA + 1;            ANA0220
           COMPACTA_REDUZIDA = SIM;                                      ANA0221
      END;                                                               ANA0222
      ELSE COMPACTA_REDUZIDA = NAO;                                      ANA0223
-/* COMPACTAR LINHA ORIGINAL E GRAVAR (4.4)                             */ANA0224
0     CALL COMPACTAR_LINHA(ORIGINAL, LINHA_ORIGINAL, INDEX_ORIGINAL,     ANA0225
      N_ORIGINAL, K);                                                    ANA0226
-/* COMPACTAR LINHA REDUZIDA E GRAVAR (4.5)                             */ANA0227
0     IF COMPACTA_REDUZIDA = SIM                                         ANA0228
      THEN CALL COMPACTAR_LINHA(REDUZIDA, LINHA_REDUZIDA, INDEX_RED,     ANA0229
           N_REDUZIDA, ULTIMA_LINHA_REDUZIDA );                          ANA0230
      ELSE;                                                              ANA0231
-/* GRAVAR LINHAS LINHAS ORIGINAL E REDUZIDA (4.6)                      */ANA0232
      CALL GRAVAR_LINHA(ORIGINAL);                                       ANA0233
      IF COMPACTA_REDUZIDA = SIM                                         ANA0234
      THEN CALL GRAVAR_LINHA(REDUZIDA);                                  ANA0235
```

```
     ELSE;                                                              ANA0236
 END;   /* FIM DO K = 1 ... */                                         ANA0237
1/* CRIAR INDEXADOR DA REDUZIDA COMPACTADA (4.7)                      */ANA0238
0DO I = 1 TO NUMERO_DE_LINHAS;                                         ANA0239
     K = REDUTOR_DE_LINHAS(I);                                         ANA0240
     INDEX_REDUZIDA(I) = INDEX_RED(K);                                 ANA0241
 END;                                                                  ANA0242
 NUMERO_PARES_REDUZIDA_COMP =                                          ANA0243
                          INDEX_RED(NUMERO_LINHAS_DA_REDUZIDA + 1);ANA0244
-/* GRAVAR INDEXADORES DAS MATRIZES (4.8)                             */ANA0245
0NUMERO_PARES_ORIGINAL_COMP =                                          ANA0246
                        INDEX_ORIGINAL(NUMERO_DE_LINHAS + 1);ANA0247
 CALL GRAVAR_REGISTRO(NAO);                                            ANA0248
 N = 0;                                                                ANA0249
 J = 1;                                                                ANA0250
 CALL GRAVAR_CHAR(NUMERO_LINHAS_DA_REDUZIDA, REDUTOR_DE_LINHAS);       ANA0251
 CALL GRAVAR_CHAR(NUMERO_COLUNAS_DA_REDUZIDA, REDUTOR_DE_COLUNAS);     ANA0252
 N = 80;                                                               ANA0253
 CALL GRAVAR_CHAR(UM, REDUTOR_DE_COLUNAS);                             ANA0254
 J = 2;                                                                ANA0255
 N = 0;                                                                ANA0256
 I = INDEX_ORIGINAL(NUMERO_DE_LINHAS);                                 ANA0257
 CALL GRAVAR_BINFIXED(I, INDEX_ORIGINAL);                              ANA0258
 N = 40;                                                               ANA0259
 CALL GRAVAR_BINFIXED(UM, INDEX_ORIGINAL);                             ANA0260
 J = 3;                                                                ANA0261
 N = 0;                                                                ANA0262
 I = INDEX_REDUZIDA(NUMERO_LINHAS_DA_REDUZIDA);                        ANA0263
 CALL GRAVAR_B  IFIXED(I, INDEX_REDUZIDA);                             ANA0264
 N = 2 * N;                                                            ANA0265
 CALL GRAVAR_C  R(NUMERO_COLUNAS_DA_REDUZIDA, REDUTOR_DE_COLUNAS);     ANA0266
 N = 80;                                                               ANA0267
 CALL GRAVAR_CHAR(UM, REDUTOR_DE_COLUNAS);                             ANA0268
1/*************************************************************/ANA0269
 /******************** COMPACTAR LINHA (4.9) ********************/ANA0270
 /*************************************************************/ANA0271
0COMPACTAR_LINHA:                                                      ANA0272
 PROC(MATRIZ, LINHA, INDEX, N, NUMERO_DA_LINHA);                       ANA0273
0DCL MATRIZ                          CHAR(1),                          ANA0274
     LINHA(256)                      CHAR(1),                          ANA0275
     INDEX(257)                      BIN FIXED,                        ANA0276
     N                               BIN FIXED,                        ANA0277
     NUMERO_DA_LINHA                 BIN FIXED;                        ANA0278
0DCL INICIO                          BIN FIXED,                        ANA0279
     FIM                             BIN FIXED,                        ANA0280
     QUEBRA                          CHAR(1),                          ANA0281
     (Y, J)                          BIN FIXED,                        ANA0282
     YCARAC(2)                       CHAR(1) BASED(ADDR(Y)),           ANA0283
     BIT2(4)                         BIT(2) BASED(ADDR(QUEBRA));       ANA0284
0IF MATRIZ = ORIGINAL                                                  ANA0285
 THEN INICIO = INICIO_DA_LINHA;                                        ANA0286
 ELSE INICIO = 1;                                                      ANA0287
 IF MATRIZ = ORIGINAL                                                  ANA0288
 THEN FIM = TAMANHO_DA_LINHA;                                          ANA0289
 ELSE FIM = NUMERO_COLUNAS_DA_REDUZIDA;                                ANA0290
 QUEBRA = LINHA(INICIO);                                               ANA0291
 J = INDEX(NUMERO_DA_LINHA);                                           ANA0292
 INDEX(NUMERO_DA_LINHA) = INDEX(NUMERO_DA_LINHA) + 1;                  ANA0293
 DO Y = INICIO TO FIM;                                                 ANA0294
      IF LINHA(Y) = QUEBRA                                             ANA0295
```

```
     THEN;                                                                  ANA0296
     ELSE                                                                   ANA0297
     DO;                                                                    ANA0298
          CALL GRAVAR_MATRIZ;                                               ANA0299
          QUEBRA = LINHA(Y);                                                ANA0300
          J = J + 1;                                                        ANA0301
     END;                                                                   ANA0302
 END;                                                                       ANA0303
 QUEBRA = LINHA(FIM);                                                       ANA0304
 Y = FIM + 1;                                                               ANA0305
 CALL GRAVAR_MATRIZ;                                                        ANA0306
 INDEX(NUMERO_DA_LINHA + 1) = J + 1;                                        ANA0307
1/*************************************************************************/ANA0308
 /********************** GRAVAR MATRIZ (4.9.1) ***************************/ANA0309
 /*************************************************************************/ANA0310
OGRAVAR_MATRIZ:                                                             ANA0311
 PROC;                                                                      ANA0312
 N = N + 1;                                                                 ANA0313
 IF N > 64                                                                  ANA0314
 THEN                                                                       ANA0315
 DO;                                                                        ANA0316
     IF MATRIZ = ORIGINAL                                                   ANA0317
     THEN WRITE FILE(ARQCP1) FROM (REG_ORIGINAL);                           ANA0318
     ELSE WRITE FILE(ARQCP2) FROM (REG_REDUZIDA);                           ANA0319
     N = 1;                                                                 ANA0320
 END;                                                                       ANA0321
 ELSE;                                                                      ANA0322
 Y = Y - 1;                                                                 ANA0323
 IF MATRIZ = ORIGINAL                                                       ANA0324
 THEN                                                                       ANA0325
 DO;                                                                        ANA0326
     PAR_ORIGINAL.A(N) = YCARAC(2);                                         ANA0327
     PAR_ORIGINAL.B(N) = BIT2(4);                                           ANA0328
 END;                                                                       ANA0329
 ELSE                                                                       ANA0330
 DO;                                                                        ANA0331
     PAR_REDUZIDA.A(N) = YCARAC(2);                                         ANA0332
     PAR_REDUZIDA.B(N) = BIT2(4);                                           ANA0333
 END;                                                                       ANA0334
 Y = Y + 1;                                                                 ANA0335
OEND;   /* FIM DE GRAVAR_MATRIZ */                                          ANA0336
OEND;   /* FIM DE COMPACTAR_LINHA */                                        ANA0337
1/*************************************************************************/ANA0338
 /********************** GRAVAR LINHA (4.6.1) ****************************/ANA0339
 /*************************************************************************/ANA0340
OGRAVAR_LINHA:                                                              ANA0341
 PROC(MATRIZ);                                                              ANA0342
ODCL MATRIZ                          CHAR(1);                               ANA0343
ODCL LINHA(256)                      CHAR(1) BASED(L),                      ANA0344
     VETOR(320)                      BIT(2) BASED(Q),                       ANA0345
     (L, Q)                          POINTER,                               ANA0346
     INICIO                          BIN FIXED INIT(1),                     ANA0347
     FIM                             BIN FIXED,                             ANA0348
     CARACTER                        CHAR(1),                               ANA0349
     BIT2(4)                         BIT(2) BASED(ADDR(CARACTER)),          ANA0350
     (K, N)                          BIN FIXED INIT(0);                     ANA0351
OFIM = TAMANHO_DA_LINHA;                                                    ANA0352
 IF MATRIZ = ORIGINAL                                                       ANA0353
 THEN                                                                       ANA0354
 DO;                                                                        ANA0355
```

```
     INICIO = INICIO_DA_LINHA;                                          ANA0356
     Q = ADDR(REG_MAT_ORIGINAL);                                        ANA0357
     L = ADDR(LINHA_ORIGINAL);                                          ANA0358
 END;                                                                   ANA0359
 ELSE                                                                   ANA0360
 DO;                                                                    ANA0361
     FIM = NUMERO_COLUNAS_DA_REDUZIDA;                                  ANA0362
     Q = ADDR(REG_MAT_REDUZIDA);                                        ANA0363
     L = ADDR(LINHA_REDUZIDA);                                          ANA0364
 END;                                                                   ANA0365
 DO K = INICIO TO FIM;                                                  ANA0366
     CARACTER = LINHA(K);                                               ANA0367
     N = N + 1;                                                         ANA0368
     IF N > 320                                                         ANA0369
     THEN                                                               ANA0370
     DO;                                                                ANA0371
          IF MATRIZ = ORIGINAL                                          ANA0372
          THEN WRITE FILE (ARQPREC) FROM (REG_MAT_ORIGINAL);            ANA0373
          ELSE WRITE FILE (ARQRED) FROM (REG_MAT_REDUZIDA);             ANA0374
          N = 1;                                                        ANA0375
     END;                                                               ANA0376
     ELSE;                                                              ANA0377
     VETOR(N) = BIT2(4);                                                ANA0378
 END;                                                                   ANA0379
OEND;   /* FIM DE GRAVAR_LINHA */                                       ANA0380
1/*****************************************************************/ANA0381
 /********************* GRAVAR BINFIXED (4.8.1) ********************/ANA0382
 /*****************************************************************/ANA0383
OGRAVAR_BINFIXED:                                                       ANA0384
 PROC(TAMANHO_DO_INDEXADOR, INDEXADOR);                                 ANA0385
ODCL TAMANHO_DO_INDEXADOR              BIN FIXED,                       ANA0386
     INDEXADOR(*)                      BIN FIXED,                       ANA0387
     K                                 BIN FIXED;                       ANA0388
ODO K = 1 TO TAMANHO_DO_INDEXADOR;                                      ANA0389
     N = N + 1;                                                         ANA0390
     IF N > 40                                                          ANA0391
     THEN                                                               ANA0392
     DO;                                                                ANA0393
          IF J = 2                                                      ANA0394
          THEN WRITE FILE (ARQCP1) FROM (REG_ORIGINAL);                 ANA0395
          ELSE WRITE FILE (ARQCP2) FROM (REG_ORIGINAL);                 ANA0396
          N = 1;                                                        ANA0397
     END;                                                               ANA0398
     ELSE;                                                              ANA0399
     VETOR_BINFIXED(N) = INDEXADOR(K);                                  ANA0400
 END;                                                                   ANA0401
OEND;   /* FIM DE GRAVAR_BIN FIXED */                                   ANA0402
1/*****************************************************************/ANA0403
 /********************* GRAVAR CHAR (4.8.2) ************************/ANA0404
 /*****************************************************************/ANA0405
OGRAVAR_CHAR:                                                           ANA0406
 PROC(TAMANHO_DO_INDEXADOR, INDEXADOR);                                 ANA0407
ODCL TAMANHO_DO_INDEXADOR              BIN FIXED,                       ANA0408
     INDEXADOR(*)                      BIN FIXED;                       ANA0409
ODCL PALAVRA                           BIN FIXED,                       ANA0410
     PALAVRA_CARAC(2)                  CHAR(1) BASED(ADDR(PALAVRA)),    ANA0411
     K                                 BIN FIXED;                       ANA0412
ODO K = 1 TO TAMANHO_DO_INDEXADOR;                                      ANA0413
     PALAVRA = INDEXADOR(K);                                            ANA0414
     N = N + 1;                                                         ANA0415
```

```
     IF N > 80                                                          ANA0416
     THEN                                                               ANA0417
     DO;                                                                ANA0418
          IF J = 1                                                      ANA0419
          THEN WRITE FILE (ARQRED) FROM (REG_ORIGINAL);                 ANA0420
          ELSE WRITE FILE (ARQCP1) FROM (REG_ORIGINAL);                 ANA0421
          N = 1;                                                        ANA0422
     END;                                                               ANA0423
     ELSE;                                                              ANA0424
     VETOR_CHAR(N) = PALAVRA_CARAC(2);                                  ANA0425
 END;                                                                   ANA0426
OEND;   /* FIM DE GRAVAR_CHAR */                                        ANA0427
1/*****************************************************************/ANA0428
 /****************** GRAVAR REGISTRO (4.8.3) **************************/ANA0429
 /*********************************************************************/ANA0430
OGRAVAR_REGISTRO:                                                       ANA0431
 PROC(REGISTRO_CONTROLE);                                               ANA0432
ODCL REGISTRO_CONTROLE                  CHAR(1);                        ANA0433
OIF REGISTRO_CONTROLE = SIM                                             ANA0434
 THEN                                                                   ANA0435
 DO;                                                                    ANA0436
     REG_MAT_ORIGINAL = CONTROLE;                                       ANA0437
     REG_MAT_REDUZIDA = CONTROLE;                                       ANA0438
     REG_ORIGINAL = CONTROLE;                                           ANA0439
     REG_REDUZIDA = CONTROLE;                                           ANA0440
 END;                                                                   ANA0441
 ELSE;                                                                  ANA0442
 WRITE FILE (ARQPREC) FROM (REG_MAT_ORIGINAL);                          ANA0443
 WRITE FILE (ARQRED) FROM (REG_MAT_REDUZIDA);                           ANA0444
 WRITE FILE (ARQCP1) FROM (REG_ORIGINAL);                               ANA0445
 WRITE FILE (ARQCP2) FROM (REG_REDUZIDA);                               ANA0446
OEND;   /* FIM DE GRAVAR_REGISTRO */                                    ANA0447
OEND;   /* FIM DO BLOCO BEGIN */                                        ANA0448
1/*****************************************************************/ANA0449
 /********************** OBTER LINHA (5) ******************************/ANA0450
 /*********************************************************************/ANA0451
OOBTER_LINHA:                                                           ANA0452
 PROC(NUMERO_DA_LINHA);                                                 ANA0453
ODCL NUMERO_DA_LINHA                    BIN FIXED;                      ANA0454
ODCL CHAVE                              PIC '(8)9',                     ANA0455
     CHAVE_DEC                          DEC FIXED(3),                   ANA0456
     PRECEDENCIA_FRACA                  BIN FIXED INIT(1),              ANA0457
     MENOR_BIT          ALIGNED         BIT(8) INIT('00000001'B),       ANA0458
     IGUAL_BIT          ALIGNED         BIT(8) INIT('00000010'B),       ANA0459
     MENOR_IGUAL_BIT    ALIGNED         BIT(8) INIT('00000100'B),       ANA0460
     MENOR                              CHAR(1) BASED(ADDR(MENOR_BIT)), ANA0461
     IGUAL                              CHAR(1) BASED(ADDR(IGUAL_BIT)), ANA0462
     MENOR_IGUAL                        CHAR(1)                         ANA0463
                                        BASED(ADDR(MENOR_IGUAL_BIT)),   ANA0464
     K                                  BIN FIXED;                      ANA0465
OCHAVE_DEC = NUMERO_DA_LINHA - 1;                                       ANA0466
 CHAVE = CHAVE_DEC;                                                     ANA0467
 READ FILE (ARQMAT) INTO (LINHA_LIDA) KEY (CHAVE);                      ANA0468
 DO K = 1 TO INICIO_DA_LINHA - 1;                                       ANA0469
      COLUNA(K) = MENOR;                                                ANA0470
 END;                                                                   ANA0471
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0472
 THEN                                                                   ANA0473
 DO K = INICIO_DA_LINHA TO TAMANHO_DA_LINHA;                            ANA0474
      IF COLUNA(K) = IGUAL  OR COLUNA(K) = MENOR_IGUAL                  ANA0475
```

```
      THEN COLUNA(K) = MENOR;                                          ANA0476
      ELSE;                                                            ANA0477
 END;                                                                  ANA0478
 ELSE;                                                                 ANA0479
OEND;   /* FIM DE OBTER_LINHA */                                       ANA0480
1/*****************************************************************/ANA0481
 /********************* IMPRIMIR RELATORIO (6) ********************/ANA0482
 /*****************************************************************/ANA0483
OBEGIN;                                                               ANA0484
ODCL PAGINA                          BIN FIXED INIT(0),               ANA0485
     CONTROLES(40)                   BIN FIXED BASED                  ANA0486
                                     (ADDR(CONTROLE)),                ANA0487
     NUMERO_DE_TERMINAIS             BIN FIXED,                       ANA0488
     NUMERO_DE_PRODUCOES             BIN FIXED,                       ANA0489
     NAOTERMINAIS                    BIN FIXED,                       ANA0490
     TAMANHO_DOS_ARGUMENTOS          BIN FIXED,                       ANA0491
     PRECEDENCIA                     CHAR(7),                         ANA0492
     (I, N, Q)                       FLOAT(15),                       ANA0493
     (J, K)                          BIN FIXED;                       ANA0494
-/* DECLARACAO DAS CONSTANTES E FUNCOES                              */ANA0495
ODCL PRECEDENCIA_FRACA               BIN FIXED INIT(1),               ANA0496
     ANAGRAMA                        CHAR(22) INIT(                   ANA0497
                          'A  N  A  G  R  A  M  A'),                  ANA0498
     PAG                             CHAR(5) INIT('PAG. '),           ANA0499
     ANALISADOR                      CHAR(43),                        ANA0500
     RELATORIO_DOIS                  CHAR(28) VAR INIT(''),           ANA0501
     ASTERISCO                       CHAR(116) VAR INIT(''),          ANA0502
     CABECALHO                       CHAR(60) VAR INIT(''),           ANA0503
     RESULTADOS                      CHAR(28) INIT(                   ANA0504
                                     'RESULTADOS DA COMPACTACAO DA'), ANA0505
     MATRIZ_DE_PRECEDENCIA           CHAR(22) INIT(                   ANA0506
                                     ' MATRIZ DE PRECEDENCIA'),       ANA0507
     ANALISA                         CHAR(24) INIT(                   ANA0508
                                     'ANALISADOR DE GRAMATICAS'),     ANA0509
     DE_PRECEDENCIA                  CHAR(15) INIT(                   ANA0510
                                     ' DE PRECEDENCIA'),              ANA0511
     RELATORIO                       CHAR(19) INIT(                   ANA0512
                                     'RELATORIO 03   -   '),          ANA0513
     DATE                            BUILTIN,                         ANA0514
     ADDR                            BUILTIN,                         ANA0515
     DATA_PLI                        CHAR(6),                         ANA0516
     ANO_MES_DIA(3)                  CHAR(2) BASED                    ANA0517
                                     (ADDR(DATA_PLI)),                ANA0518
     MES                             PIC'99',                         ANA0519
     DATA                            CHAR(9) VAR INIT(''),            ANA0520
     MESES_ANO(12)                   CHAR(3) INIT(                    ANA0521
                         'JAN','FEV','MAR','ABR','MAI','JUN',         ANA0522
                         'JUL','AGO','SET','OUT','NOV','DEZ'),        ANA0523
     TRACO                           CHAR(1) INIT('-'),               ANA0524
     FRACA                           CHAR(7) INIT('FRACA  '),         ANA0525
     SIMPLES                         CHAR(7) INIT('SIMPLES'),         ANA0526
     SIM                             CHAR(1) INIT('S');               ANA0527
-/* FORMATAR E IMPRIMIR CABECALHO (6.1)                              */ANA0528
ODO K = 1 TO 116;                                                     ANA0529
      ASTERISCO = ASTERISCO CAT '*';                                  ANA0530
 END;                                                                 ANA0531
 ANALISADOR = ANALISA CAT DE_PRECEDENCIA;                             ANA0532
 DATA_PLI = DATE;                                                     ANA0533
 MES = ANO_MES_DIA(2);                                                ANA0534
 DATA = ANO_MES_DIA(3) CAT TRACO CAT MESES_ANO(MES)                   ANA0535
```

```
       CAT TRACO CAT ANO_MES_DIA(1);                                    ANA0536
 RELATORIO_DOIS = RELATORIO CAT DATA;                                   ANA0537
 CABECALHO = RESULTADOS CAT MATRIZ_DE_PRECEDENCIA;                      ANA0538
 PAGINA = 1;                                                            ANA0539
 PUT EDIT(ANAGRAMA, PAG, PAGINA, ANALISADOR, RELATORIO_DOIS)            ANA0540
 (PAGE,SKIP,COL(50),A,COL(108),A,F(3),SKIP(2),COL(41),A,SKIP(2),        ANA0541
 COL(47),A);                                                            ANA0542
 PUT EDIT(ASTERISCO,'*',CABECALHO,'*',ASTERISCO)                        ANA0543
 (SKIP(3),COL(5),A,COL(5),A,COL(35),A,COL(120),A,COL(5),A);             ANA0544
-/* IMPRIMIR CARACTERISTICAS DA GRAMATICA (6.2)                        */ANA0545
ONUMERO_DE_PRODUCOES = CONTROLES(4);                                    ANA0546
 NAOTERMINAIS = CONTROLES(2);                                           ANA0547
 TAMANHO_DOS_ARGUMENTOS = CONTROLES(5);                                 ANA0548
 NUMERO_DE_TERMINAIS = NUMERO_DE_LINHAS - NAOTERMINAIS;                 ANA0549
 IF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0550
 THEN PRECEDENCIA = FRACA;                                              ANA0551
 ELSE PRECEDENCIA = SIMPLES;                                            ANA0552
 PUT EDIT('CARACTERISTICAS DA GRAMATICA',                               ANA0553
          '--------------------------',                                 ANA0554
          '- NUMERO DE PRODUCOES: ', NUMERO_DE_PRODUCOES,               ANA0555
          '- NUMERO DE TERMINAIS: ', NUMERO_DE_TERMINAIS,               ANA0556
          '- NUMERO DE NAO TERMINAIS: ', NAOTERMINAIS,                  ANA0557
          '- TAMANHO ACUMULADO DOS ARGUMENTOS: ',TAMANHO_DOS_ARGUMENTOS,ANA0558
          '- TIPO DE PRECEDENCIA: ',PRECEDENCIA)                        ANA0559
 (SKIP(5), COL(26), A, SKIP, COL(26), A,                                ANA0560
 4(SKIP(2),COL(31),A,F(4)),                                             ANA0561
 SKIP(2),COL(31),A,A);                                                  ANA0562
1/* IMPRIMIR RESULTADOS (6.3)                                          */ANA0563
OIF INDICADOR_DE_PRECEDENCIA = PRECEDENCIA_FRACA                        ANA0564
 THEN J = NUMERO_DE_LINHAS - INICIO_DA_LINHA + 1;                       ANA0565
 ELSE J = NUMERO_DE_LINHAS;                                             ANA0566
 N = J * NUMERO_DE_LINHAS / 4  + 1;                                     ANA0567
 PUT EDIT('RESULTADOS DA COMPACTACAO',                                  ANA0568
          '--------------------------',                                 ANA0569
 'MATRIZ ORIGINAL : ', NUMERO_DE_LINHAS, ' LINHAS E ', J, ' COLUNAS,',  ANA0570
 ' OCUPANDO ', N, ' CARACTERES.')                                       ANA0571
 (SKIP(5), COL(26),A,SKIP,COL(26),A,                                    ANA0572
 SKIP(2), COL(10),A,F(3),A,F(3),A,A,F(5),A);                            ANA0573
0Q = NUMERO_LINHAS_DA_REDUZIDA * NUMERO_COLUNAS_DA_REDUZIDA;            ANA0574
 Q = Q / 4 + 1;                                                         ANA0575
 Q = Q + NUMERO_LINHAS_DA_REDUZIDA + NUMERO_COLUNAS_DA_REDUZIDA;        ANA0576
 I = 100 * (1 - Q / N);                                                 ANA0577
 PUT EDIT ('MATRIZ REDUZIDA : ',NUMERO_LINHAS_DA_REDUZIDA, ' LINHAS, ', ANA0578
 NUMERO_COLUNAS_DA_REDUZIDA,' COLUNAS, DOIS VETORES INDEXADORES ',      ANA0579
 ' DE LINHAS E COLUNAS,','OCUPANDO TUDO ', Q, ' CARACTERES, COM ',      ANA0580
 I, '% DE COMPACTACAO.')                                                ANA0581
 (SKIP(2),COL(10),A,F(3),A,F(3),A,A,SKIP,COL(10),A,F(5),A,F(7,2),A);    ANA0582
0J = NUMERO_PARES_ORIGINAL_COMP;                                        ANA0583
 Q = 1.25 * J + 2 * NUMERO_DE_LINHAS + 1;                               ANA0584
 I = 100 * (1 - Q / N);                                                 ANA0585
 PUT EDIT('MATRIZ ORIGINAL COMPACTADA : ', J,' PARES (Y, PRECEDENCIA), 'ANA0586
 ,NUMERO_DE_LINHAS, ' PONTEIROS DE LINHA,','OCUPANDO TUDO ',            ANA0587
 Q, ' CARACTERES, COM ', I,'% DE COMPACTACAO.')                         ANA0588
 (SKIP(2),COL(10),A,F(5),A,F(3),A,SKIP,COL(10),A,F(5),A,F(7,2),A);      ANA0589
```

```
0 J = NUMERO_PARES_REDUZIDA_COMP;                                          ANA0590
  Q = 1.25 * J + 1;                                                        ANA0591
  Q = Q + 2 * NUMERO_DE_LINHAS + NUMERO_COLUNAS_DA_REDUZIDA;               ANA0592
  I = 100 * (1 - Q / N);                                                   ANA0593
  PUT EDIT('MATRIZ REDUZIDA COMPACTADA : ',J,' PARES (Y, PRECEDENCIA),',   ANA0594
  NUMERO_LINHAS_DA_REDUZIDA, ' PONTEIROS DE LINHA, ',                      ANA0595
  NUMERO_COLUNAS_DA_REDUZIDA, ' INDEXADORES DE ',                          ANA0596
 'COLUNAS, OCUPANDO TUDO ',Q, ' CARACTERES, COM ', I,                      ANA0597
 '% DE COMPACTACAO.')                                                      ANA0598
 (SKIP(2),COL(10),A,                                                       ANA0599
 F(5),A,F(3),A,F(3),A,SKIP,COL(10),A,F(5),A,F(7,2),A);                     ANA0600
0END;   /* FIM DO BLOCO BEGIN */                                           ANA0601
0END;   /* FIM DE ANAG301 */                                               ANA0602
```

Por outro lado o desenvolvimento do trabalho, ao multiplicar os casos de ajuda mútua e atividade comum e ao mostrar assim a cada indivíduo as vantagens desta colaboração, contribuía forçosamente a agrupar ainda mais os membros da sociedade. Em resumo, os homens no seu processo de formação chegaram a um ponto que tiveram a necessidade de dizer algo uns aos outros. A necessidade criou o órgão: a laringe pouco desenvolvida do macaco foi se transformando lenta e firmemente produzindo modulações cada vez mais diferenciadas, enquanto a boca aprendia pouco a pouco a pronunciar uma palavra após outra.

"Humanização do Macaco pelo Trabalho"

F. ENGELS

CAPÍTULO VII

CONCLUSÕES

É possível sugerir reparos na concepção do ANAGRAMA e a primeira delas talvez se refira ao tipo de gramática escolhida. Realmente, as gramáticas de precedência às vezes limitam as linguagens projetadas. No entanto, é preciso lembrar os objetivos didáticos apresentados no Capítulo I. As gramáticas de precedência são extremamente apropriadas à compreensão da teoria das linguagens formais, constituindo portanto uma introdução adequada aos demais tipos de gramáticas livres do contexto. Além do mais, um reconhecedor sintático voltado para linguagens de precedência é, sem dúvida, mais simples e mais econômico. Se uma determinada aplicação é perfeitamente atendida por uma linguagem de precedência, por que não usá-la?

Um outro aspecto do ANAGRAMA, talvez o principal, é a sua filosofia de automatização e agilização a ser seguida nos projetos de linguagens e reconhecedores, partindo-se apenas das produções da gramática. Esta talvez seja a idéia básica da proposta. A disseminação de pacotes semelhantes ao apresentado é oportuna e útil e o próprio ANAGRAMA pode ser modificado ou ampliado de modo a incorporar outros tipos de gramáticas livres do contexto.

Aceitas e conservadas as idéias básicas do pacote, alguns melhoramentos podem, no entanto, serem prontamente implementados no esquema de programação utilizado. Por exemplo, seria interessante o usuário poder interferir nas entradas dos Módulos 02 e 03, modificando o tipo de precedência para os ca-

sos em que ele quisesse tratar precedência simples como precedência fraca. Outra idéia seria o Relatório 01 ser facultativo ou poder ser restringido a algumas de suas partes quando isto fosse desejável. A redução dos números de arquivos físicos intermediários gerados pelo Módulo 01 poderia ser tentada e isto, se fosse conseguido, seria saudável para o desempenho global do sistema. É preciso assinalar também que, na implementação do reconhecedor sintático, o projetista precisa recorrer ao Relatório 03 para saber as dimensões da matriz reduzida ou o número de pares gerados nas compactações. Seria conveniente gravar estes dados nos registros de controle dos arquivos criados para o reconhecedor, de modo que este ficasse independente das diversas versões das compactações geradas pelo ANAGRAMA. Vale notar que estas modificações, bem como acréscimos ou conversões para outra linguagem, provavelmente deverão ser feitas com relativa facilidade, em virtude das técnicas de documentação e de desenvolvimento modular/hierárquico que foram adotadas.

Enfim, o ANAGRAMA pretende apenas ser uma proposta inicial. É esperado que o resultado final dos melhoramentos introduzidos pelo seu uso consiga atingir os resultados definidos e inspirar outros sistemas semelhantes, mais eficientes e úteis aos projetos de compiladores.

ANEXO I

O desenvolvimento e a documentação do ANAGRAMA foram feitos baseados em uma técnica proposta pela IBM que, na literatura original, é denominada de HIPO (Hierarchy Input Process Output). Como o próprio nome sugere, esta técnica tenta descrever um sistema como uma árvore de processos. Um processo é associado a uma função a ser desempenhada em algum nível do sistema e ele é implementado como o próprio sistema, um programa, uma rotina ou conjunto de comandos. Para cada função são definidas as entradas e as saídas. São usados dois diagramas e o entendimento aprofundado do uso e conceitos do HIPO são apresentados na bibliografia 10.

1 - Diagrama de Funções

Este diagrama é uma visualização gráfica da árvore de processos (ou funções). As entradas e saídas de cada função não são mostradas. A Figura III.2 é um exemplo deste diagrama. É convencionado o seguinte:

- um nó da árvore é desenhado como um retângulo; ali consta o título descritivo do processo e, a partir de algum nível, a numeração hierárquica colocada no canto inferior direito;
- um processo é controlado pelo processo pai; em termos de implementação, se o processo é uma rotina isto significa que ela é chamada apenas pela rotina de nível imediatamente superior; se o processo é controlado pelos processos irmãos, o retângulo será cortado por uma

barra, como no processo 6.6 da Figura V.1;

- se o processo é implementado como um programa o nome do programas vem ao lado da numeração do retângulo, como na Figura V.1; se o processo é implementado como 'procedure' ou um bloco 'begin' isto é assinalado com dois asteriscos ao lado da numeração hierarquica.

2 - Diagrama de Detalhamento

Este diagrama se refere a uma função específica. Deste modo, cada retângulo do Diagrama de Funções pode gerar um Diagrama de Detalhamento. Para a função sendo descrita são mostradas as sub-funções constituintes e para cada uma destas sub-funções são definidas as entradas e saídas. A Figura V.6 é um exemplo deste diagrama.

É convencionado o seguinte:

- as entradas e saídas são desenhadas de modo a indicar o meio onde o dado é gravado e assim são usadas as convenções gráficas habituais para disco, fita, relatório, etc.;
- as ligações lógicas entre dados e funções são feitas através de setas cheias e pontilhadas; as setas cheias indicam dados efetivos de entrada e saída da função, enquanto que a pontilhada indica que a função será executada ou não conforme o conteúdo do dado; na Figura V.6 a função 'Gerar Valores das Funções Lineares' será executada dependendo do conteúdo de 'Indicador de Ciclos'.

- o título das funções constituintes da função sendo detalhada poderá vir dentro de um retângulo ou vir acompanhada de uma nota explicativa; se vier dentro de um retângulo, isto indica que há um diagrama específico de detalhamento para aquela função; na Figura V.6 a função 'Verificar Ciclos' é descrita no diagrama da Figura V.7, enquanto a descrição de 'Verificar Ciclos' é descrita no diagrama da Figura V.7, enquanto a descrição de 'Construir Grupamentos' termina com a nota explicativa.

ANEXO II

UM EXEMPLO DE APLICAÇÃO DO ANAGRAMA

Deseja-se projetar uma gramática de precedência simples ou fraca, que descreva uma linguagem que contenha as frases:

O HOMEM CRIA O GATO

O HOMEM CRIA UM GATO

O HOMEM TEM UM GATO

O HOMEM TEM O GATO

É exigido apenas que estas frases não iniciem por 'UM'. As produções são então codificadas de modo a serem entradas para o Módulo 01. Note-se que a gramática proposta, além de gerar as frases pedidas, também gera frases como 'O GATO CRIA UM HOMEM', mas isto não será considerado restrição. Esta codificação vai listada na página seguinte. É emitido então o Relatório 01 que analisa estas produções. Como indicam as páginas 4 e 6 deste relatório, a gramática tem conflitos do tipo 'menor/maior'. É preciso então modificar as produções.

As produções são novamente submetidas ao Módulo 01 e é obtida outra versão do Relatório 01. As modificações introduzidas foram as seguintes:

- o não-terminal <artigo>> foi suprimido;
- a definição de <objeto > foi alterada.

A gramática torna-se então de precedência simples. O Módulo 02 é aplicado e, como mostra o Relatório 02, existem funções lineares de precedência. As diversas formas de compactação para a matriz de precedência são finalmente apresentadas no Relatório 03, como resultado da aplicação do Módulo 03.

```
          ********************************************************
          *  EXEMPLO DE CODIFICACAO DAS PRODUCOES DE ENTRADA  *
          ********************************************************




*
*         GRAMATICA TESTE
*        (VERSAO PRIMEIRA)
*
*
<SENTENCA> <SUJEITO>          <PREDICATO>
<SUJEITO>    O  <NOME>
<PREDICATO>      <VERBO>    <OBJETO>
<OBJETO>    <ARTIGO>     <NOME>
<VERBO>    TEM
 CRIA
<ARTIGO>    UM
                      O
<NOME>      HOMEM
          GATO
```

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

****
* SEQUENCIAL * PRODUCOES DA GRAMATICA *
****

GRAMATICA TESTE

(VERSAO PRIMEIRA)

```
001     <SENTENCA>  ::=  <SUJEITO>  <PREDICATO>
002     <SUJEITO>  ::=  O  <NOME>
003     <PREDICATO>  ::=  <VERBO>  <OBJETO>
004     <OBJETO>  ::=  <ARTIGO>  <NOME>
005     <VERBO>  ::=  TEM
006                |  CRIA
007     <ARTIGO>  ::=  UM
008                |  O
009     <NOME>  ::=  HOMEM
010                |  GATO
```

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SEQUENCIAL	RELACAO DOS NAO-TERMINAIS	PRODUCOES ONDE O SIMBOLO OCORRE
001	<SENTENCA>	1
002	<SUJEITO>	1, 2
003	<PREDICATO>	1, 3
004	<OBJETO>	3, 4
005	<VERBO>	3, 5, 6
006	<ARTIGO>	4, 7, 8
007	<NOME>	2, 4, 9, 10

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SEQUENCIAL	RELACAO DOS TERMINAIS	PRODUCOES ONDE O SIMBOLO OCORRE
008	O	2, 8
009	TEM	5
010	CRIA	6
011	UM	7
012	HOMEM	9
013	GATO	10
014	ZZ	*** SIMBOLO CRIADO COMO DELIMITADOR DE CADEIA ***

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SIMBOLO		VIZINHANCA A DIREITA		PRECEDENCIA NO PAR	
1	<SENTENCA>	14	ZZ	>	
2	<SUJEITO>	3	<PREDICATO>	=	
2	<SUJEITO>	5	<VERBO>	<	
2	<SUJEITO>	9	TEM	<	
2	<SUJEITO>	10	CRIA	<	
3	<PREDICATO>	14	ZZ	>	
4	<OBJETO>	14	ZZ	>	
5	<VERBO>	4	<OBJETO>	=	
5	<VERBO>	6	<ARTIGO>	<	
5	<VERBO>	8	O	<	
5	<VERBO>	11	UM	<	
6	<ARTIGO>	7	<NOME>	=	
6	<ARTIGO>	12	HOMEM	<	
6	<ARTIGO>	13	GATO	<	
7	<NOME>	9	TEM	>	
7	<NOME>	10	CRIA	>	
7	<NOME>	14	ZZ	>	
8	O	7	<NOME>	=	
8	O	12	HOMEM	><	***
8	O	13	GATO	><	***
9	TEM	8	O	>	
9	TEM	11	UM	>	
10	CRIA	8	O	>	
10	CRIA	11	UM	>	

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SIMBOLO		VIZINHANCA A DIREITA		PRECEDENCIA NO PAR
11	UM	12	HOMEM	>
11	UM	13	GATO	>
12	HOMEM	9	TEM	>
12	HOMEM	10	CRIA	>
12	HOMEM	14	ZZ	>
13	GATO	9	TEM	>
13	GATO	10	CRIA	>
13	GATO	14	ZZ	>
14	ZZ	1	<SENTENCA>	<
14	ZZ	2	<SUJEITO>	<
14	ZZ	8	O	<

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

****
* DIAGNOSTICOS E MENSAGENS DE ERRO *
****

***** A GRAMATICA NAO E' DE PRECEDENCIA
EXISTEM CONFLITOS ENVOLVENDO A RELACAO ' > '

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

* SEQUENCIAL * PRODUCOES DA GRAMATICA *

GRAMATICA TESTE

(VERSAO MODIFICADA)

```
001        <SENTENCA>  ::=  <SUJEITO>  <PREDICATO>
002        <SUJEITO>  ::=  O  <NOME>
003        <PREDICATO>  ::=  <VERBO>  <OBJETO>
004        <OBJETO>  ::=  <SUJEITO>
005                     |  UM  <NOME>
006        <VERBO>  ::=  TEM
007                     |  CRIA
008        <NOME>  ::=  HOMEM
009                     |  GATO
```

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SEQUENCIAL	RELACAO DOS NAO-TERMINAIS	PRODUCOES ONDE O SIMBOLO OCORRE
001	<SENTENCA>	1
002	<SUJEITO>	1, 2, 4
003	<PREDICATO>	1, 3
004	<OBJETO>	3, 4, 5
005	<VERBO>	3, 6, 7
006	<NOME>	2, 5, 8, 9

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

* SEQUENCIAL *	RELACAO DOS TERMINAIS	* PRODUCOES ONDE O SIMBOLO OCORRE *
007	0	2
008	UM	5
009	TEM	6
010	CRIA	7
011	HOMEM	8
012	GATO	9
013	ZZ	*** SIMBOLO CRIADO COMO DELIMITADOR DE CADEIA ***

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

	SIMBOLO		VIZINHANCA A DIREITA	PRECEDENCIA NO PAR
1	<SENTENCA>	13	ZZ	>
2	<SUJEITO>	3	<PREDICATO>	=
2	<SUJEITO>	5	<VERBO>	<
2	<SUJEITO>	9	TEM	<
2	<SUJEITO>	10	CRIA	<
2	<SUJEITO>	13	ZZ	>
3	<PREDICATO>	13	ZZ	>
4	<OBJETO>	13	ZZ	>
5	<VERBO>	2	<SUJEITO>	<
5	<VERBO>	4	<OBJETO>	=
5	<VERBO>	7	O	<
5	<VERBO>	8	UM	<
6	<NOME>	9	TEM	>
6	<NOME>	10	CRIA	>
6	<NOME>	13	ZZ	>
7	O	6	<NOME>	=
7	O	11	HOMEM	<
7	O	12	GATO	<
8	UM	6	<NOME>	=
8	UM	11	HOMEM	<
8	UM	12	GATO	<
9	TEM	7	O	>
9	TEM	8	UM	>
10	CRIA	7	O	>

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

SIMBOLO		VIZINHANCA A DIREITA		PRECEDENCIA NO PAR
10	CRIA	8	UM	>
11	HOMEM	9	TEM	>
11	HOMEM	10	CRIA	>
11	HOMEM	13	ZZ	>
12	GATO	9	TEM	>
12	GATO	10	CRIA	>
12	GATO	13	ZZ	>
13	ZZ	1	<SENTENCA>	<
13	ZZ	2	<SUJEITO>	<
13	ZZ	7	O	<

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 01 - 13-SET-79

DIAGNOSTICOS E MENSAGENS DE ERRO

A GRAMATICA E' DE PRECEDENCIA SIMPLES
- SIMBOLO INICIAL DA GRAMATICA: <SENTENCA>
- TAMANHO ACUMULADO DOS ARGUMENTOS: 13

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 02 - 13-SET-79

****
* CARACTERISTICAS DA GRAMATICA E FUNCOES LINEARES DE PRECEDENCIA *
****

CARACTERISTICAS DA GRAMATICA

- NUMERO DE PRODUCOES: 9
- NUMERO DE TERMINAIS: 7
- NUMERO DE NAO TERMINAIS: 6
- TAMANHO ACUMULADO DOS ARGUMENTOS: 13
- TIPO DE PRECEDENCIA: SIMPLES

VALORES DAS FUNCOES LINEARES

FUNCAO F :

F(1)= 1, F(2)= 1, F(3)= 1, F(4)= 1, F(5)= 0, F(6)= 3, F(7)= 0
F(8)= 0, F(9)= 2, F(10)= 2, F(11)= 3, F(12)= 3, F(13)= 0

FUNCAO G :

G(1)= 1, G(2)= 1, G(3)= 1, G(4)= 0, G(5)= 2, G(6)= 0, G(7)= 1
G(8)= 1, G(9)= 2, G(10)= 2, G(11)= 1, G(12)= 1, G(13)= 0

ANALISADOR DE GRAMATICAS DE PRECEDENCIA

RELATORIO 03 - 13-SET-79

****
* RESULTADOS DA COMPACTACAO DA MATRIZ DE PRECEDENCIA *
****

CARACTERISTICAS DA GRAMATICA

- NUMERO DE PRODUCOES: 9
- NUMERO DE TERMINAIS: 7
- NUMERO DE NAO TERMINAIS: 6
- TAMANHO ACUMULADO DOS ARGUMENTOS: 13
- TIPO DE PRECEDENCIA: SIMPLES

RESULTADOS DA COMPACTACAO

MATRIZ ORIGINAL : 13 LINHAS E 13 COLUNAS, OCUPANDO 43 CARACTERES.

MATRIZ REDUZIDA : 7 LINHAS, 11 COLUNAS, DOIS VETORES INDEXADORES DE LINHAS E COLUNAS, OCUPANDO TUDO 38 CARACTERES, COM 11.05% DE COMPACTACAO.

MATRIZ ORIGINAL COMPACTADA : 53 PARES (Y, PRECEDENCIA), 13 PONTEIROS DE LINHA, OCUPANDO TUDO 93 CARACTERES, COM -116.86% DE COMPACTACAO.

MATRIZ REDUZIDA COMPACTADA : 32 PARES (Y, PRECEDENCIA), 7 PONTEIROS DE LINHA, 11 INDEXADORES DE COLUNAS, OCUPANDO TUDO 78 CARACTERES, COM -81.40% DE COMPACTACAO.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AHO, Alfred & ULLMAN, Jeffrey D. - The Theory of Parsing, Translation and Compiling, Vols. I e II, New Jersey, Prentice-Hall, 1972.

2. BIRKHOFF, G. & BARTEE, T. C. - Modern Applied Algebra, New York, McGraw - Hill Book Company, 1970.

3. CHOMSKY, N. - On Certain Formal Properties of Grammar, Information and Control 2, 1959, pp. 137-167.

4. FELDMAN, J. & GRIES, D. - Translator Writing Systems, vol 11, nº 2, fevereiro 1966, pp. 77-113.

5. GLASS, D. - An Elementary Discussion of Compiler/Interpreter Writing, Computing Surveys, vol. 1, nº 1, março 1969.

6. GRIES, D. - Compiler Construction for Digital Computers, New York John Wiley and Sons/Inc, 1971.

7. HUNT III, H.B & SZYMANSKY, T.G. & ULLMAN, J.D. - Operations on Sparse Relations, Communications of ACM, vol 20, nº 3, Março 1977, pp. 171-176.

8. Mc KEEMAN, W. M. & HORNING, J.J. & WORTMAN, D. B. - A Compile Generator, New Jersey, Prentice-Hall, 1970.

9. MONTEIRO, L.H.J. - Elementos de Álgebra, Rio de Janeiro, Ao Livro Técnico S.A., 1969.

10. NEW YORK, IBM Corporation, GN20-3644 - HIPO - A Desing Aid and Documentation Technique, 1975

11. WEINGARTEN, F.W. - Translation of Computer Languages, S. Francisco, California, Holden-Day Inc. 1973.