

PUC

Série: Monografias em Ciência da Computação
Nº 8/77

(antiga/formerly: Monographs in Computer Science
and Computer Applications).

ESTUDOS PRELIMINARES PARA ESPECIFICAÇÃO E IMPL
MENTAÇÃO DE UMA LINGUAGEM GERAL DE SIMULAÇÃO(LGS)

por

Heitor M. Quintella,
Érico de O. Silva,
Vilmondes G. Silva.

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Quês de São Vicente 225 — ZC 19

Rio de Janeiro — Brasil

Série: Monografias em Ciência da Computação
Nº 8/77

(antiga/formerly: Monographs in Computer Science
and Computer Applications).

ESTUDOS PRELIMINARES PARA ESPECIFICAÇÃO E IMPLEMEN-
TAÇÃO DE UMA LINGUAGEM GERAL DE SIMULAÇÃO (LGS)*

por

Heitor M. Quintella,
Érico de Oliveira Silva,
Vilmondes G. da Silva.

Editor: Michael F. Challis

agosto 1977

* Trabalho patrocinado em parte pela FINEP e CNPq.

RESUMO:

Nesse trabalho apresentam-se os estudos iniciais realizados no Departamento de Informática da PUC para o desenvolvimento de uma linguagem geral de simulação (LGS) e que resultaram na produção do primeiro protótipo. Nessa fase tratou-se exclusivamente de explorar as características do GPSS que estarão também presentes na LGS. Em particular estudou-se: a) a possibilidade de melhorar as estruturas de tratamento das transações por meio de uma 'simulation language base machine' (LBM) que é o conceito abstrato de cadeia.

b) a definição de uma linguagem de especificação de modelos de nível mais alto que o GPSS utilizando a forma de Jensen e Wirth.

PALAVRAS CHAVE: SIMULAÇÃO DISCRETA, LINGUAGENS DE SIMULAÇÃO ,
SISTEMAS DE FILAS.

ABSTRACT:

In this work we present a preliminary study done at the Dep. de Informática PUC for the development of a general simulation language (LGS) that allowed in the production of the first prototype. In the present phase we dealt with GPSS-like characteristics that will be present in L.G.S. In particular we studied: a) the possibility of improving some structures for transaction manipulation by means of a simulation language base machine (LBM) that is the abstract concept of chains.

b) the definition of a model specification language of a higher level than GPSS, utilizing Jensen and Wirth's form.

KEY WORDS: DISCRETE SIMULATION, SIMULATION LANGUAGES, SISTEMAS DE FILAS.

SUMÁRIO

1 - INTRODUÇÃO	1
2 - TÉCNICA GERAL PARA O CONTROLE DO FLUXO DE COMPONENTES DE UM SISTEMA	3
3 - DEFINIÇÃO FORMAL DE UM SUBCONJUNTO DA LGS	6
4 - UM EXEMPLO EM LGS	18
5 - CONCLUSÃO	19
REFERÊNCIAS BIBLIOGRÁFICAS	20

1 - INTRODUÇÃO

As linguagens de simulação discreta tratam da interação no tempo de um número de processos inter-relacionados. Dessa forma processos podem ser modelados por meio de uma sequência discreta de eventos. Numa simulação deste tipo (orientada para eventos) o programa lida com a organização de eventos que operam sobre estruturas de dados denominadas entidades. Assim, eventos modificam valores de atributos das entidades ou as criam, ou as destroem. Além disso eles estão inter-relacionados por meio de uma série de mecanismos de acesso. Chama-se tipo-abstrato de dados de uma linguagem de simulação ao espaço de dados de eventos e o conjunto de operações definidos nesse espaço. Os eventos por sua vez operam sobre as entidades. Chama-se então máquina da linguagem de simulação (MBLS) ao conjunto de objetos abstratos estruturados que representam as entidades.

A preocupação central da construção de um compilador de linguagem de simulação discreta consiste no estabelecimento do controle do fluxo dos componentes do sistema através do mesmo. Isto envolve uma série de problemas de estrutura interna dos dados.

Em linguagens como o SIMSCRIPT [DIMSDALE, B., MARKOWITZ, A. 64] ou SIMULA 67 [DAHL, O. J. et al 68] o usuário tem até certo ponto grande liberdade de escolher seus tipos abstratos mas nunca escolhe a sua MBLS. Em GPSS ele não tem escolha sobre nenhum dos dois.

Tenta-se resolver esse inconveniente no LGS pelo uso de uma MBLS de muito alto nível que permite a representação mais geral possível para entidades e tipos abstratos.

O CONCEITO ABSTRATO DE CADEIAS COMO UMA MBLS

A maioria das linguagens de simulação discreta opera sobre uma MBLS que tem uma estrutura de cadeia. Tal conceito pode ser formalizado como segue:

Seja E um conjunto. Chama-se estado s de uma cadeia ao conjunto totalmente ordenado $E \subseteq E$. Chama-se $E(s)$ ao conjunto suporte de s , e $\bar{E}(s)$ ao seu complemento.

Seja S um conjunto de estados. Chame-se modificador m a uma aplicação de S em S , e M um conjunto de modificadores.

Uma cadeia \bar{e} é então um par (S, M) que obedece às seguintes propriedades:

$$a. \exists \{ \emptyset, s \} \in S \text{ tal que } E(s) = \emptyset$$

$$b. \exists I \in M \text{ onde } I \text{ é o modificador identidade}$$

$$c. \forall s \in S, \text{ existe uma sequência } (m_i)_{i \in \mathbb{N}}, m_i \in M \text{ tal que}$$

$$s = m_n \circ m_{n-1} \circ \dots \circ m_1(\emptyset)$$

$$d. \forall (s, m) \in S \times M, \text{ as ordens induzidas por } s \text{ e } m(s) \text{ sobre } E(s) \cap E(m(s)) \text{ são idênticas}$$

$$e. \forall (s, m) \in S \times M, E(s) \cap \bar{E}(m(s)) \text{ e } \bar{E}(s) \cap E(m(s)) \text{ são finitas.}$$

Uma estrutura de cadeias assim definida requer so mente dois modificadores fundamentais cuja semântica é a seguinte:

$$E(m(s)) = E(m^- \circ m^+(s))$$

onde:

$$m^-(s) = -E(s) \cap \bar{E}(m(s))$$

e

$$m^+(s) = \bar{E}(s) \cap E(m(s))$$

Demonstra-se [Quintella, H.M., 76] que a maior parte das estruturas de dados que ocorre em simulação pode ser tratada como um caso particular de estrutura em cadeias. O grande alcance dessa observação é que se pode construir compiladores que utilizem esse conceito como um MBLIS pelo menos de duas maneiras:

- a) - pela utilização em bruto do conceito mais geral para tratamento de todas as estruturas
- b) - pela utilização de uma forma refinada do conceito que permita uma otimização das operações em certos casos particulares de estruturas.

Na primeira versão do L.G.S. está-se estudando a aplicação da primeira alternativa como filosofia de construção do compilador por motivos de simplicidade. Nessa fase adotou-se, em particular, uma estrutura de controle de fluxo de componentes semelhante à do GPSS.

2 - TÉCNICA GERAL PARA O CONTROLE DO FLUXO DE COMPONENTES DE UM SISTEMA.

Foi implementado um programa que faz a leitura dos cartões com a codificação do modelo simulado feita na linguagem LGS.

Este programa transfere os comandos de LGS para um analisador sintático, que além da análise sintática prepara também as estruturas de dados para a execução.

A execução do processo de simulação é feita por um módulo de execução. Este módulo utiliza os conceitos de cadeia as atual e futura, fazendo a manipulação das transferências de transações indicada pelas condições da simulação. A figura 2.1 apresenta o fluxo geral do módulo de execução.

FLUXO GERAL DO SISTEMA

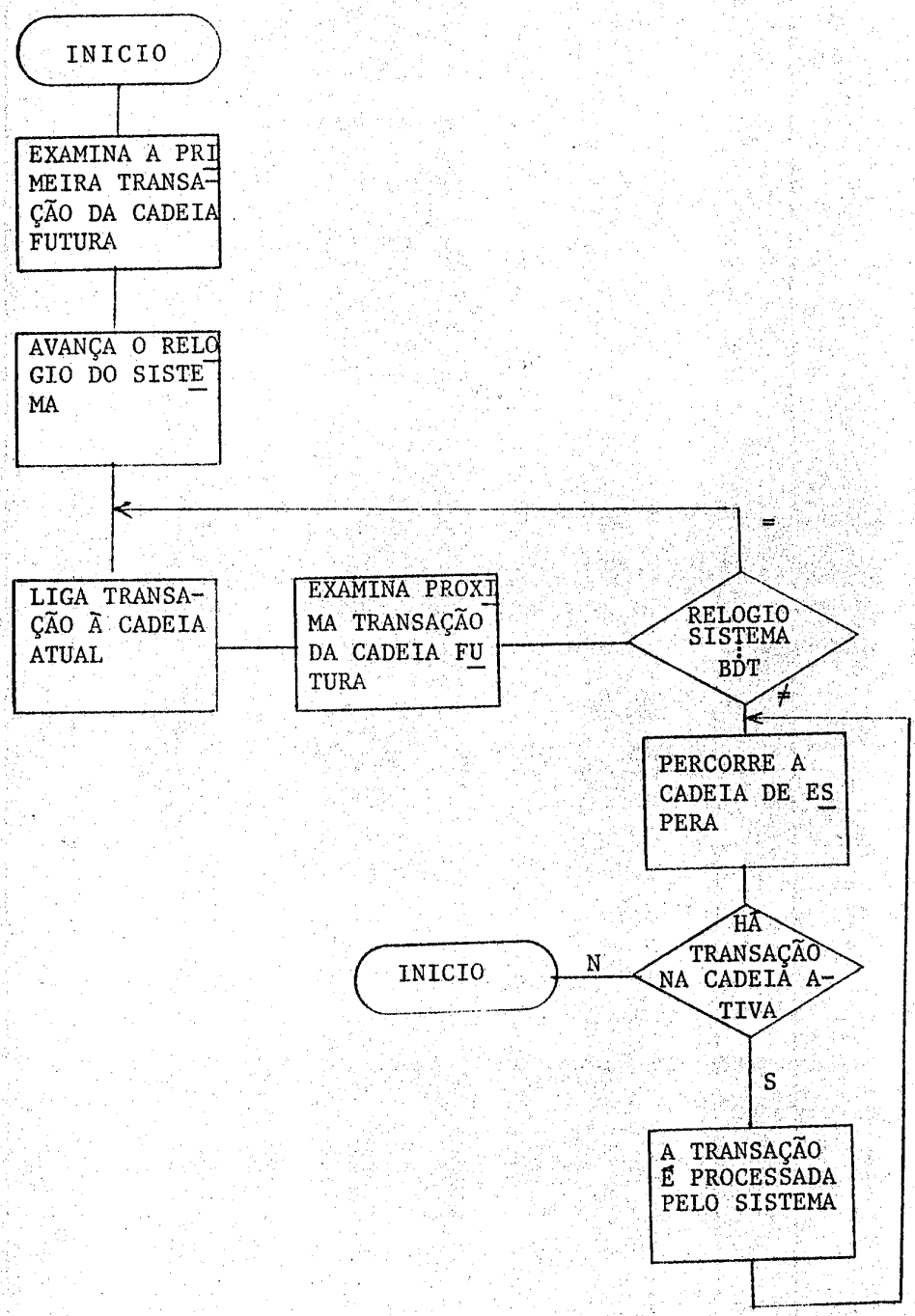


FIG. 2.1

Como apontado em [Précigout, M. 68] a cadeia a atual contém transação cujo tempo (chamado "block departure time" - BDT) é menor ou igual ao relógio do sistema em determinado instante. Na cadeia atual podemos ter transações em estado ativo, isto é, que podem movimentar-se pelo sistema, constituindo a chamada cadeia ativa, e transações em estado de espera, que não podem se movimentar pelo sistema, pelo fato de estarem esperando desocupar um determinado bloco (e.g.: um canal de serviço), constituindo a cadeia de espera.

Na cadeia futura são colocadas as transações cujo tempo (BDT) seja maior que o relógio do sistema naquele instante.

Para a manipulação dessas cadeias foram criadas subrotinas primitivas para inserção e exclusão de elementos que transitam pelas cadeias. Neste sentido foram implementadas duas estruturas diferentes de cadeias, porém cada uma de las tem de residir isoladamente no sistema, isto é, existe uma versão do sistema para cada uma delas.

Uma das estruturas utilizou o enfoque tradicional, no qual são montadas as estruturas de filas e pilhas usuais, onde cada nodo da lista que representa a cadeia possui um campo contendo informações da transação e um campo contendo o endereço (apontador) do próximo nodo, conforme Fig. 2.2. A outra estrutura conforme [Gordon, G. 62] utilizou o enfoque dos modificadores, onde é usada uma lista duplamente encadeada visando simular listas e pilhas.

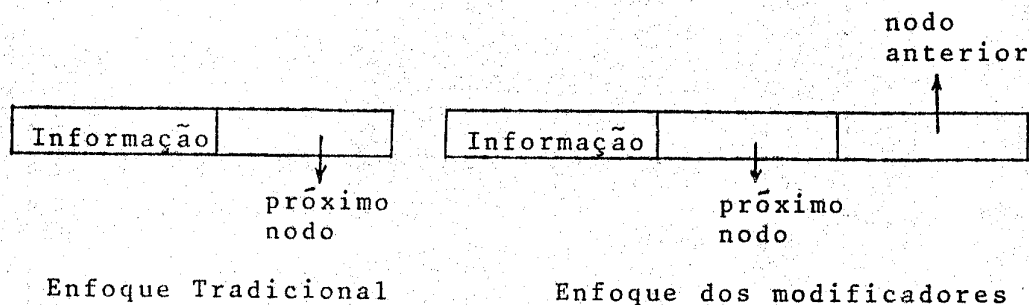


FIG. 2.2

3 - DEFINIÇÃO FORMAL DE UM SUBCONJUNTO DA LGS

Está sendo desenvolvida na PUC uma linguagem de simulação (LGS) de alto nível objetivando uma maior flexibilidade e facilidade de compreensão e utilização por parte do usuário. Neste trabalho discute-se apenas um subconjunto da LGS semelhante ao GPSS. A principal característica desta linguagem é ser de formato livre com palavras de uso corrente em português, onde os parâmetros da simulação são identificados por palavras chaves esclarecedoras.

Para que o modelo de simulação possa ser expresso num português quase natural, são permitidas inclusive variações do tipo singular/plural, por exemplo, podemos escrever:

POR TRANSACÃO NA FILA (F1);

para que uma transação entre na fila de nome F1 ou:

POR (m) TRANSACÕES NA FILA (F1);

para que m transações (m > 1) entrem na fila. É permitido também o uso de comando comentário para fornecer textos explicativos.

Cada comando pode ser perfurado nas colunas 1 a 80 do cartão e pode ocupar qualquer número de cartões. A única restrição, claramente mostrada pela sintaxe, é que não pode haver quebra nos literais e parâmetros.

3.1 - DESCRIÇÃO SINTÁTICA DA LINGUAGEM:

A descrição da sintaxe da linguagem é feita baseando-se na forma empregada por Jensen e Wirth [Jensen, K. e Wirth, N. 74]. Inicialmente serão fornecidos os conceitos básicos necessários à descrição da sintaxe.

- CONCEITOS BÁSICOS:

→ indica o sentido do caminho na descrição da sintaxe.

—| indica o fim do caminhamento



significa que se seguirá em frente obrigatoriamente se esta for a N-ésima vez que o cruzamento é atingido, caso contrário, segue-se o desvio.



significa que se pode passar no máximo N vezes por esta "ponte"

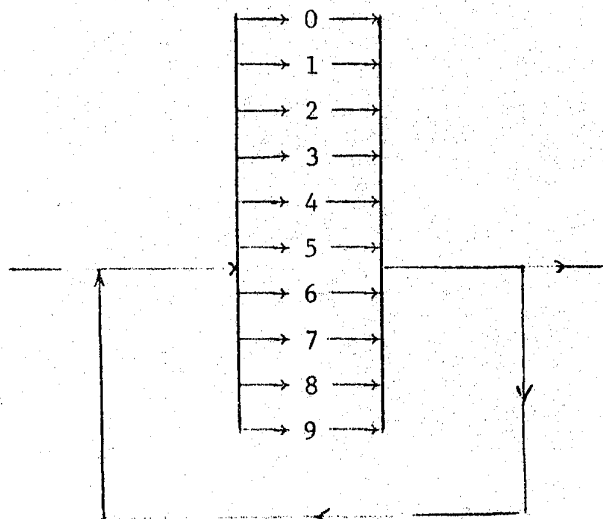
Palavras em letras minúsculas tem o seu sentido usual.

Palavras em letras maiúsculas são consideradas como sequência de caracteres.

Palavras sublinhadas são consideradas como nome de uma entidade sintática pré-definida.

3.2 - EXEMPLOS DE DESCRIÇÕES SINTÁTICAS:

3.2.1 - Definição de um número inteiro:



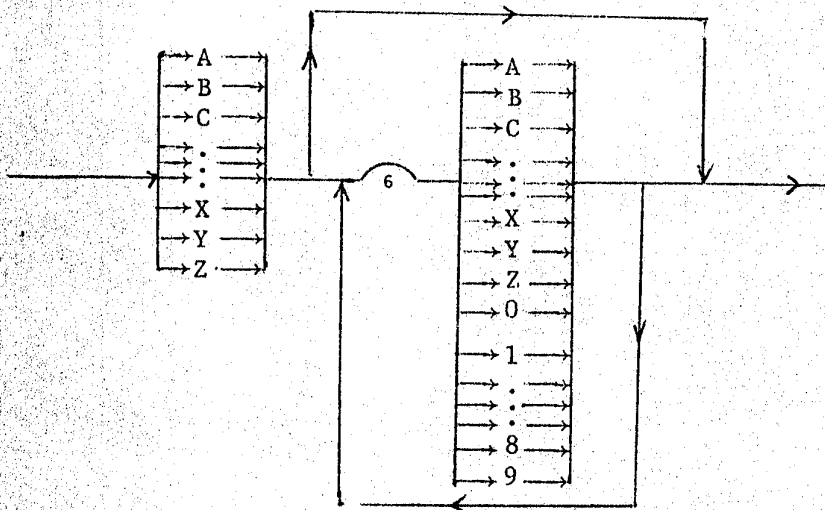
segundo esta descrição são números inteiros válidos:

0
123
173543218910835
99

e são inválidos:

4.
3,0
8.0

3.2.2 Definição de identificador que tenha 7 caracteres no máximo, sendo que o primeiro é necessariamente uma letra e os restantes podem ser letras ou dígitos.



são identificadores válidos, segundo a definição:

X
AA
A1932A0
ZZZXXXZ

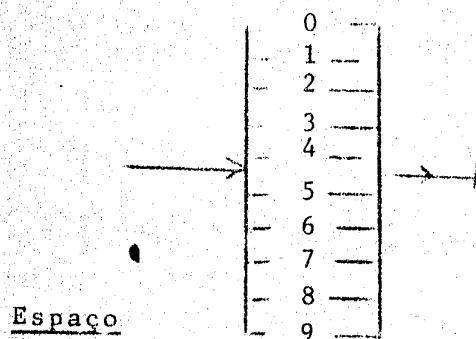
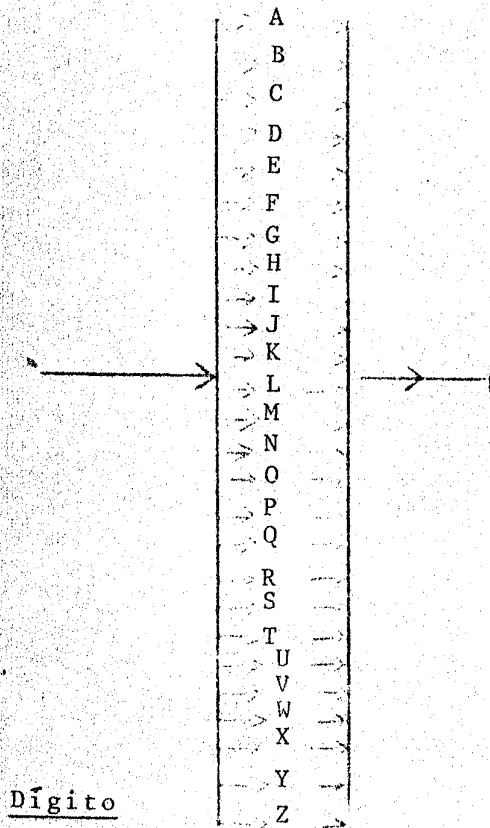
e são inválidos:

3
ABCDEXZF
AB# 32
@7

3.3 - ENTIDADES SINTÁTICAS BÁSICAS:

Algumas entidades sintáticas básicas serão estabelecidas aqui para facilitar a visualização e compreensão de entidades mais complexas.

Letra

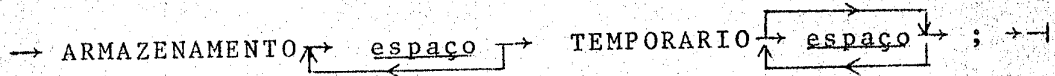


→ 1 coluna em branco →

3.4 - COMANDOS DA L G S

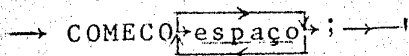
Os comandos são apresentados em ordem alfabética, com a respectiva semântica.

3.4.1 - ARMAZENAMENTO:



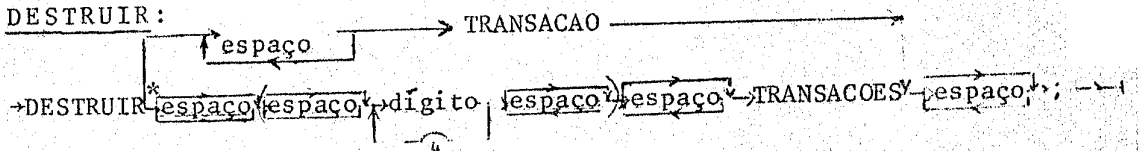
Este comando faz com que cada transação passe pela fila e pelo canal de serviço 2 vezes.

3.4.2 - COMEÇO:



O comando não é executável, apenas indica o início dos comandos do programa.

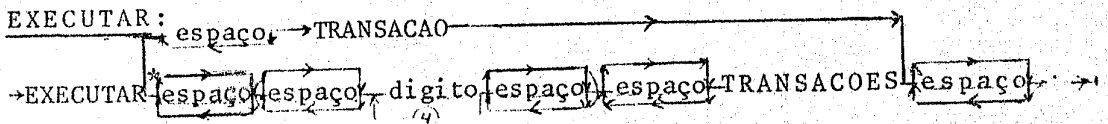
3.4.3 - DESTRUIR:



* segue-se esta opção quando o nº de transações é maior que 1.

Destroi a transação, devolve a área ocupada pela transação ao espaço disponível e atualiza os indicadores do sistema.

3.4.4 - EXECUTAR:



* segue-se esta opção quando o nº de transações é maior que 1.

Determina o nº de transações que vão percorrer todo o processo de simulação.

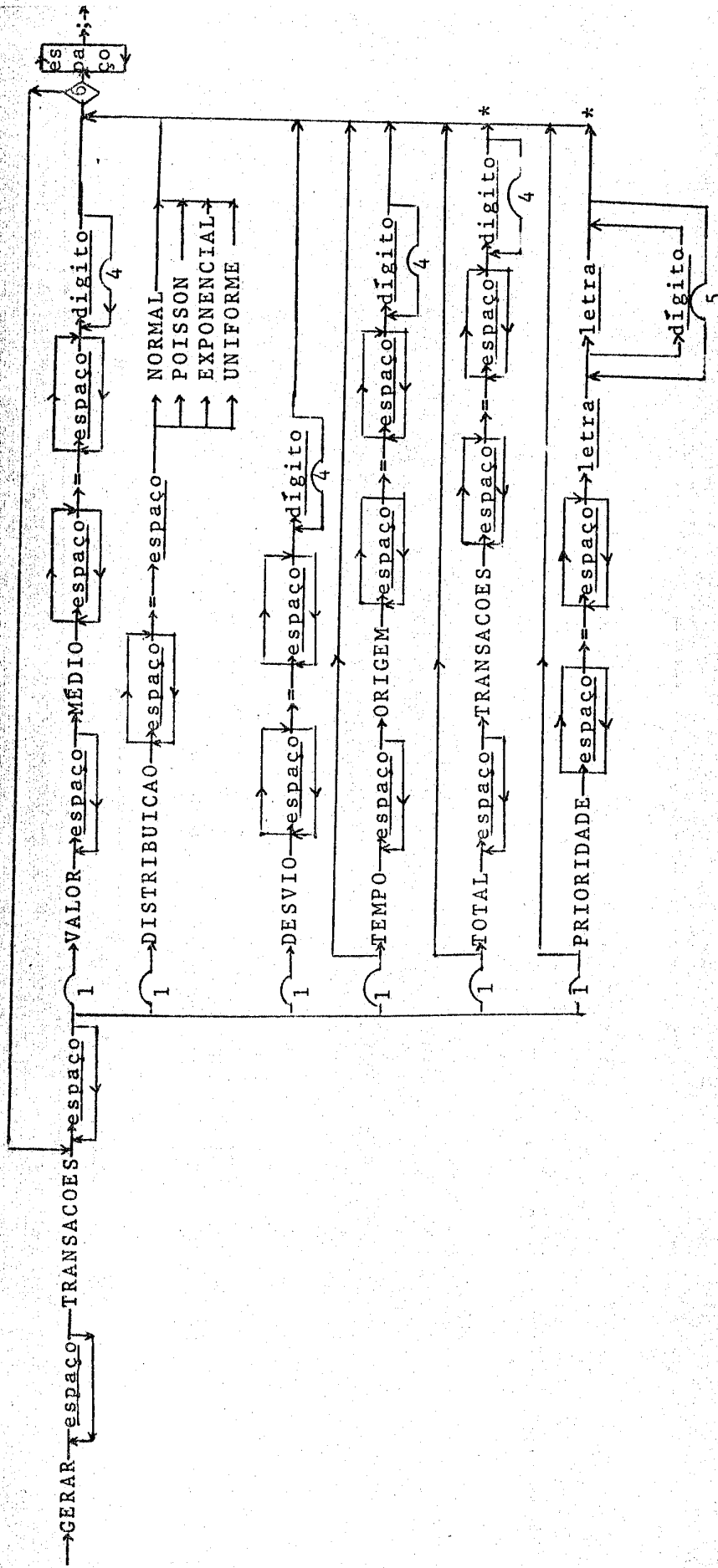
Ex.: EXECUTAR (300) TRANSACOES;
EXECUTAR TRANSACAO;

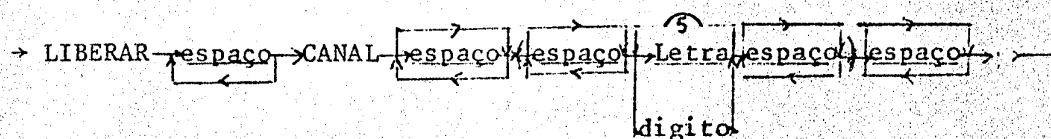
3.4.5. FIM:

— FIM — → espaço → ; — → |

Este comando não é executável, apenas indica o fim dos comandos do programa.

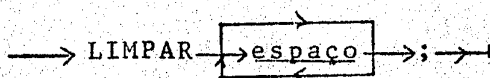
3.4.6) - GERAR:



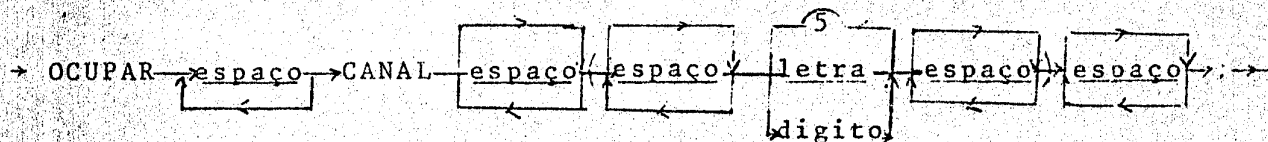
3.4.8 - LIBERAR:

Torna o canal disponível para a próxima transação e atualiza as estatísticas do canal.

Ex.: LIBERAR CANAL (CAN01)

3.4.9 - LIMPAR:

Este comando coloca em todos os indicadores do sistema o valor zero e devolve as transações ao espaço disponível.

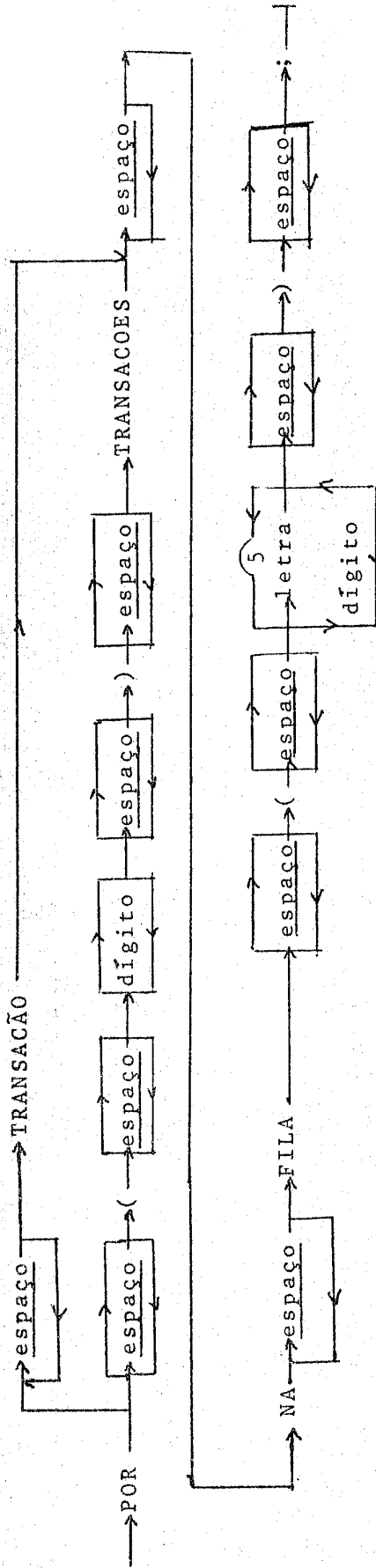
3.4.10 - OCUPAR:

Este comando aloca o canal de serviço a uma transação, se o canal está desocupado, caso contrário, a transação fica na fila de espera para o canal.

Ex.:

OCUPAR CANAL(CAN17);

3.4.11. POR:



* Segue-se esta opção quando o nº de transações for maior que 1.

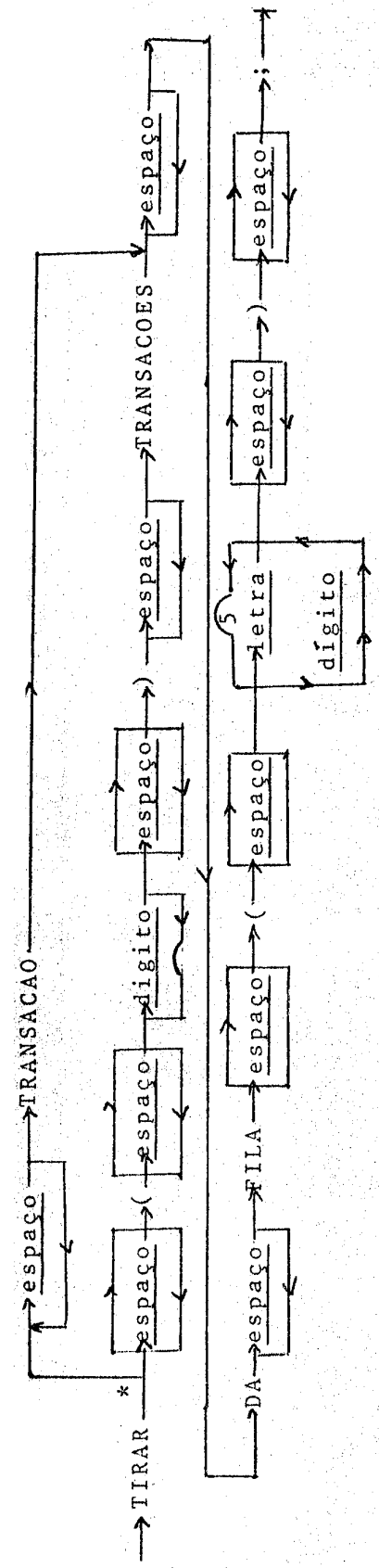
Coloca-se a(s) transação(ões) na fila indicada e atualiza os indicadores.

Ex:

POR TRANSAÇÃO NA FILA (FILAS);

POR (4) TRANSAÇÕES NA FILA (22);

3.4.12. TIRAR:



* Segue-se esta opção quando o nº de transações é maior que 1.

Retira a(s) transação(oes) da fila indicada e atualiza os indicadores da fila

Ex:

- TIRAR TRANSAÇÃO DA FILA (FILA2);
- TIRAR (2) TRANSAÇÕES DA FILA (257);

3.4.13. COMENTÁRIO

→ # → qualquer sequencia de caracteres exceto o
→ # →

O comando comentário permite explicações adicionais sobre o programa.

Ex.: # CANAL DE SERVIÇO QUE
SIMULA A
BILHETERIA. #

4 - UM EXEMPLO EM L.G.S.

Para exemplificar a utilização da linguagem, considera-se um sistema de bilheteria de cinema com um único guichê. O sistema se comporta do seguinte modo:

A chegada de pessoas à bilheteria ocorre com um intervalo médio de 10 segundos e desvio padrão de 3 segundos:

- a pessoa entra na fila
- a pessoa chega à bilheteria
- a pessoa sai da fila
- a pessoa ocupa a bilheteria durante o intervalo médio de 12 segundos, com desvio padrão de 4 segundos.
- a pessoa libera a bilheteria
- a pessoa entra no cinema

O programa para simular a situação acima, poderá ser:

```

COMÉÇO;                                     # INICIO DO EXEMPLO #
GERA TRANSACOES                             # GERA A CHEGADA DE PESSOAS#
    VALOR MEDIO = 10
    DISTRIBUICAO=UNIFORME
    DESVIO = 3;

ENTRA TRANSACAO NA FILA(FILA1);             # A PESSOA ENTRA NA FILA #
OCUPA CANAL (GUICHE);                       # A PESSOA CHEGA A BILHETERIA #
TIRA TRANSACAO DA FILA(FILA1);             # SAI DA FILA #
INCREMENTA RELOGIO                          # OCUPA A BILHETERIA #
    DISTRIBUICAO = UNIFORME
    VALOR MEDIO = 12
    DESVIO = 4;

LIBERA CANAL (GUICHE);                      # LIBERA A BILHETERIA #
DESTROI (1) TRANSACAO;                     # ENTRA NO CINEMA#
EXECUTA (500) TRANSACOES                   # SIMULA COM 500 PESSOAS #
FIM;

```

5 - CONCLUSÃO:

Pelo que foi exposto até agora pode-se depreender que:

- a. pela flexibilidade de escolha de tipos abstratos e de MBLS
- b. pela possibilidade de representação geral de estruturas abstratas
- c. pela possibilidade de estímulo a programação estruturada em simulação

é de grande interesse o desenvolvimento de uma linguagem como o L.G.S.

Em [QUINTELLA, H.M., 76] esboçou-se esquematicamente o plano do projeto de desenvolvimento da L.G.S. A essa altura do seu desenvolvimento já se pode apresentar um planejamento mais detalhado como o da figura 5.1

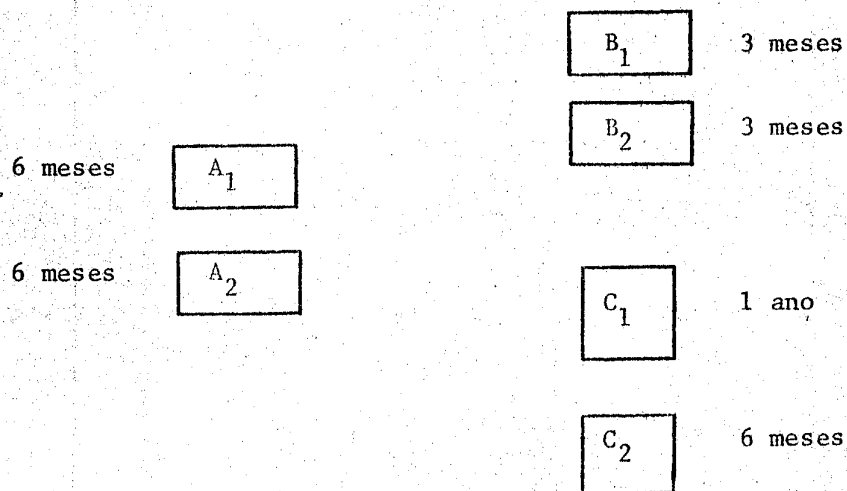


FIG. 5.1

- A₁. Definição das principais características da L.G.S.
 A₂. Definição final da linguagem.
 B₁. Representação de estruturas de dados e tipos de dados
 B₂. Mecanismos para seleção automática de estruturas de dados

- C₁. Definição da estrutura do compilador
 C₂. Implementação do compilador.

A fase B₁ está completa e o relatório dos seus resultados pode ser encontrada em [Lucena, et al, 76].
 A fase A₁ será desenvolvida por dois alunos deste departamento resultando em teses de mestrado.

As demais fases deverão ser desenvolvidas em termos menos acadêmicos podendo envolver inclusive financiamento e gerência externos à Universidade

6 - REFERÊNCIAS:

- 1 - DAHL, O. J., MYHRHAUG, NYGAARD K., Simula 67 Common Base Language, Oslo Norwegian Computer Centre, 1968.
- 2 - DI M SDALE, B., MARKOVITZ, A., A description of the Simgcript Language, IBM Systems Journal, Vol. 3 nº 1, 1964
- 3 - GORDON, G. A General purpose systems simulator, IBM Systems Journal I, 1962
- 4 - LUCENA, C. J., QUINTELLA, H. M., Schwabe, D. "On the modelling and representation of abstractions in Computer simulation languages" IFIP Working Conference on Modelling of Environmental Systems, Tokyo, Japan, April 1976
- 5 - QUINTELLA, H. M. - Fundamentos de Teoria de Cadeias para uso em simulação. Nota técnica da PUC/RJ (a ser publicado).
- 6 - JENSEN K. e WIRTH N., Pascal User Manual and Report. Lecture notes in Computer Science, Vol. 18, Springer Verlag, 1974
- 7 - PRECIGOUT, M., Um Système de Simulation: GPSS. In: Agard J. (ed), Les Methods de Simulation GPSS: Association Française D'Informatique et de Recherche Operationelle, 1968

- 8 - QUINTELLA, H.M., L.G.S. Uma alternativa para modelagem de levantamentos estatísticos. Reunião Internacional de Estatística Aplicada, São Paulo 1976.

NOTA PÓS - IMPRESSÃO

Encontrava-se em fase de desenvolvimento sob minha orientação durante a confecção deste trabalho a tese: "Estudo Comparativo de duas Linguagens de Simulação Contínua: CSMP/360 e DYNAMO II", de Diego Santimateo Galvez, que trata de estruturas contínuas de linguagens de simulação. Essa tese juntamente com esse trabalho completam a fase A1 (fig. 5-1).

Futuramente será feita uma alteração sugerida pelo Prof. Gaston Gonnet que consiste em introduzir a descrição da linguagem por meio do Backus-Naur-Formalism. Como primeiro relatório adotou-se a descrição utilizada por Niklaur Wirth por seu impacto visual.

Heitor M. Quintella