

UNIVERSIDADE ESTATUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA

ABRIL DE 1989

*Este exemplar inclui
as sugestões indicadas
pelo banca examinadora.
H. H.
27/6/89*

SISTEMA ESPECIALISTA EM MANOBRAS
DE REDES DE DISTRIBUIÇÃO
DE ENERGIA ELÉTRICA

por: Celso Cavellucci

Orientador: Christiano Lyra Filho

Tese de Mestrado apresentada à Faculdade de
Engenharia Elétrica da Universidade Esta-
dual de Campinas.

UNICAMP
BIBLIOTECA CENTRAL

Eu ouço, e eu esqueço.

Eu vejo, e eu lembro.

Eu faço, e eu entendo.

(proverbio chinês)

Ao meus pais Ireny e Mercedes.

À Lia, Murilo e Bruno.

Agradeço sinceramente

ao Christiano pela confiança, apoio e tudo mais
que levou a concretização deste trabalho,

ao Sigmar pela valiosa contribuição no cálculo da
queda de tensão pelo método dos momentos,

ao Simoni pelo apoio incondicional de muitos anos.

ao Chico e ao João pela amizade e incentivo,

ao Marcos e Aurelio pela amizade e pelas valiosas
idéias e sugestões,

ao Murilo e Bruno, cuja curiosidade amenizou
muitas horas de trabalho,

a Lia pela compreensão, incentivo e o amor, que
tornou possível realizar este trabalho,

Enfim, a todas as pessoas que contribuíram, direta
ou indiretamente, para conclusão deste trabalho.

ÍNDICE

APRESENTAÇÃO	1
CAPÍTULO 1 - DESCRIÇÃO GERAL DO SISTEMA RD	3
CAPÍTULO 2 - ESTRUTURA INTERNA DO SISTEMA	10
ESPECIALISTA RD	
2.1. REPRESENTAÇÃO DO CONHECIMENTO	11
2.2. ESTRUTURA BÁSICA DE CONSULTA	16
2.3. RACIOCÍNIO POR CONTEXTO	19
2.4. ÁRVORE DE CONTEXTO DO SISTEMA RD	20
2.5. FUNÇÕES PARA AVALIAÇÃO DAS REGRAS	26
2.6. REGRAS DO SISTEMA RD	28
2.7. MECANISMO DE INFERÊNCIA	31
2.8. EXEMPLO DE UMA CONSULTA AO	36
SISTEMA RD	
2.9. RESUMO E COMENTÁRIOS	54
CAPÍTULO 3 - SIMULADOR DE MANOBRA	56
3.1. MODELO DA REDE DE DISTRIBUIÇÃO	57
DE ENERGIA ELÉTRICA	
3.2. ALGUNS RESULTADOS MATEMÁTICOS	59
3.3. CÁLCULO DA QUEDA DE TENSÃO DEVIDA	62
À VARIAÇÃO DE CARGA	
3.4. ALGORITMO DE SIMULAÇÃO DE MANOBRA	63
3.5. ESTRUTURA DE DADOS	70
3.6. ASPECTOS DE IMPLEMENTAÇÃO DA	73
ESTRUTURA DE DADOS	
CONCLUSÃO	80
APÊNDICE A	
APÊNDICE B	
REFERÊNCIAS BIBLIOGRÁFICAS	

APRESENTAÇÃO

Este trabalho propõe um sistema especialista integrado a procedimentos de computação numérica, denominado *RD* e projetado para sugerir alternativas de planos de manobra no isolamento e restabelecimento de redes de distribuição de energia elétrica. Os planos de manobra são baseados no conhecimento do especialista (operador de rede) e em avaliações das condições elétricas da manobra.

O conhecimento do especialista é representado por fatos na forma de triplas <contexto, atributo, valor> e regras de produção, que são avaliadas convenientemente pela máquina de inferência do sistema *RD*. Sua concepção está baseada na experiência do sistema *MYCIN*, desenvolvido na universidade de Stanford.

A rede de distribuição está representada por um modelo de grafo, onde os nós são blocos de carga e os arcos chaves do sistema, armazenados numa estrutura de dados orientada por arco.

A avaliação dos requisitos elétricos (queda de tensão e capacidade física dos componentes da rede) é obtida através de um simulador de manobra (computação numérica). O simulador de manobra baseia-se no Método dos Momentos para o cálculo da queda de tensão. Este método foi adaptado para utilizar algoritmos construídos a partir de conceitos de grafos, o que resultou numa abordagem extremamente

eficiente.

Com base nestas idéias foi desenvolvido um protótipo do sistema RD. Os módulos referentes à computação simbólica foram implementados em PROLOG e os módulos de computação numérica em linguagem C.

O capítulo 1 apresenta a arquitetura básica de sistema especialista adotada neste trabalho. Em seguida, descreve a arquitetura específica do sistema especialista RD.

O capítulo 2 explora a estrutura interna do sistema especialista RD no que diz respeito à representação do conhecimento e do mecanismo de inferência. Um exemplo comentado de uma consulta é apresentado, mostrando passo a passo o encadeamento do raciocínio realizado pela máquina de inferência.

O capítulo 3 aborda os aspectos referentes aos módulos de computação numérica. Descreve o modelo de grafo que representa a rede de distribuição, apresenta o algoritmo para simulação de manobra e os aspectos da estrutura de dados adotada.

CAPÍTULO 1

DESCRIÇÃO GERAL DO SISTEMA RD

O gerenciamento da rede de distribuição compreende, entre outras, as funções de monitoração e controle. Entende-se por monitoração as informações para avaliar as condições da rede e por controle as ações para alterar essas condições. As ações, na maioria das vezes, implicam em uma ou mais manobras na rede de distribuição - sequência de abrir e fechar chaves.

As manobras podem ocorrer devido a interrupções não programadas, momentâneas, ou ainda por interrupções programadas para atender as necessidades de manutenção da rede e de transferência de carga entre alimentadores e subestações. Tanto as interrupções não programadas (emergência) como as programadas implicam em manobras de isolamento e restabelecimento da rede. A sequência de operações na rede, que define a manobra a ser realizada, deve obedecer requisitos elétricos e critérios previamente estabelecidos, normatizados por órgãos governamentais ou pela concessionária de distribuição de energia elétrica. O conjunto de procedimentos adotados para a execução da manobra denomina-se *Plano de Manobra*.

Os operadores de rede são responsáveis pela execução dos planos de manobra. Nesta tarefa, eles tomam

constantemente decisões sobre como isolar uma subestação, como isolar um alimentador de distribuição, como realocar carga entre alimentadores, como isolar uma parte da rede com defeito e, posteriormente, como restabelecer o fornecimento de energia nas áreas afetadas da rede de distribuição (denominada *área escura*). Tanto o isolamento como o restabelecimento exigem análise detalhada de um grande número de possibilidades de manobra, que buscam uma alternativa que satisfaça requisitos elétricos, exigências de segurança, critérios sociais e normas da empresa. A experiência do operador é fundamental na execução de planos de manobra, sejam eles emergenciais ou programados.

Muitos problemas de engenharia elétrica admitem formulação analítica rigorosa na forma de um modelo matemático e podem ser resolvidos através de algoritmos. Os algoritmos são transformados em programas computacionais que encontram as soluções dos problemas. Existe também outra classe de problemas cujas soluções dependem do bom senso, experiências anteriormente adquiridas no desempenho da função e do julgamento próprio de engenheiros ou técnicos (os especialistas).

Como considerar os aspectos empíricos nos modelos matemáticos? Duas abordagens são utilizáveis. Na primeira, acrescenta-se heurística à formulação matemática como elemento restritivo de soluções indesejadas, tornando assim o modelo mais "inteligente" [CAS1, CAST1]. Outra abordagem,

mais recente, emprega técnicas de representação e tratamento do conhecimento, objeto de estudo da inteligência artificial (IA) [WOL1]. Em particular, na resolução do problema de restabelecimento de redes elétricas algumas técnicas de IA são empregadas: busca heurísticas [HARR1, MOR1], sistema baseado em conhecimento [SAK1] e sistema especialista [LIU1].

Os SE são programas computacionais que resolvem problemas em uma área específica e limitada do conhecimento humano, imitando modelos de raciocínio e inferência características de especialistas na área. Estes sistemas utilizam a representação explícita do conhecimento, nomeada *Base de Conhecimento*. A figura 1.1. esquematiza o arquitetura básica de um SE.

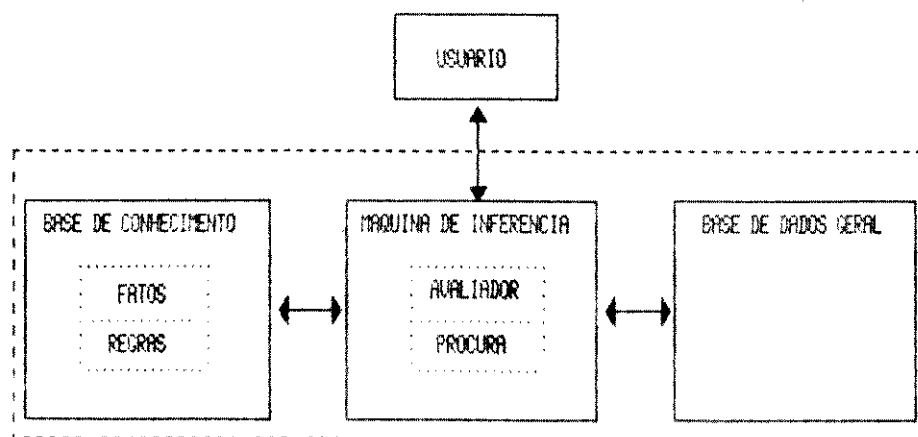


FIGURA 1.1. Arquitetura básica de um Sistema Especialista.

A base de conhecimento contém os FATOS

(informações consideradas verdadeiras) e *REGRAS* (critérios comprovados teoricamente, de bom senso ou suposições aceitáveis de uma determinada área). A *Máquina de Inferência* avalia as regras adequando-as aos fatos armazenados conforme a estratégia de controle adotada. A *BASE DE DADOS GLOBAL*, algumas vezes chamada de "Quadro Negro", é uma memória de trabalho cuja função principal é armazenar o estado atual do sistema [SCH1].

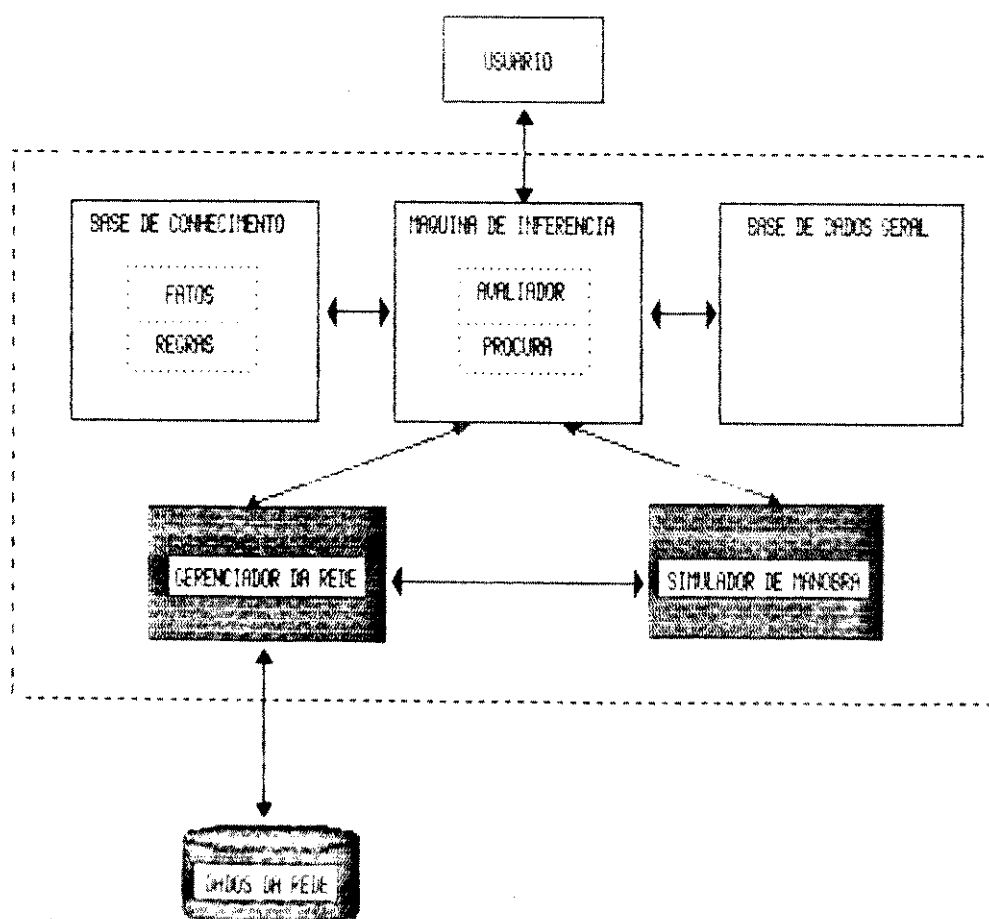


FIGURA 1.2. Arquitetura do RD.

A arquitetura específica do RD, um sistema especialista para auxílio de manobras em redes de distribuição, está representada esquematicamente na figura 1.2.

A base de conhecimento deste sistema compreende fatos e regras obtidas a partir da experiência dos operadores de rede, representando leis de "consenso". Os fatos são armazenados na forma de triplas <contexto, atributo, valor>, onde o contexto são objetos tratados pelas regras do RD (rede, bloco com defeito, alimentador, manobra e chave), definidos pelos respectivos parâmetros (atributos), que podem assumir determinados valores. As regras são implementadas como *regras de produção* [RIC1] do tipo *SE premissa ENTÃO ação* tratadas convenientemente pela máquina de inferência.

A estratégia de inferência usada no RD é do tipo "SE a premissa é verdadeira ENTÃO execute a ação" (*Modus Ponens*). O controle de inferência é dirigido por metas em encadeamento para trás (backward chaining) e o raciocínio suportado é monotônico, isto é, todos os fatos inferidos permanecem inalterados até o fim de uma consulta ao sistema.

A máquina de inferência é composta por dois módulos interrelacionados: *AVALIADOR* e *PROCURA* [BUC1]. O *AVALIADOR* analisa cada condição da premissa da regra e se todas as condições forem verdadeiras executa a ação. Caso contrário, a regra é rejeitada.

Se o AVALIADOR não dispuser de todos os dados necessários para satisfazer a condição da regra, ele aciona o módulo PROCURA. Este módulo busca encontrar os valores dos contextos e dos atributos associados ao contexto, para que o AVALIADOR continue sua análise. O mecanismo de busca do PROCURA pode levar este módulo a solicitar informações ao usuário, acionar um dos módulos de apoio (blocos com fundo escuro na figura 1.2), ou ainda inferir novos fatos por meio de regras a serem analisadas pelo AVALIADOR (Observe o processo recursivo AVALIADOR $\rightleftarrows$ PROCURA).

O *Gerenciador da Rede* e o *Simulador de Manobra* representam os componentes responsáveis pela computação numérica do sistema RD. O armazenamento, tratamento e atualização da estrutura de dados que representa a rede de distribuição é realizado pelo Gerenciador da Rede. O Simulador de Manobra verifica se a queda de tensão e a capacidade de corrente nos elementos da rede (cabos, chaves, ...) afetados pela manobra estão dentro de limites pré-estabelecidos. Como resultado, o simulador indica a adequação ou não da manobra pretendida. Um algoritmo procedimental implementa este módulo, usando um modelo de grafo onde os nós representam os blocos de carga e subestações e os arcos representam os dispositivos seccionadores (chave) da rede.

O arquivo *Dados da Rede* contém informações da topologia da rede de distribuição e parâmetros elétricos

necessários ao cálculo de queda de tensão e da capacidade física dos elementos de rede. A topologia da rede é armazenada como uma estrutura de dados de grafo comumente denominada *Orientada por Arcos* [JEN1]. A rede primária em operação, é armazenada por uma árvore em *preordem*. A estrutura em preordem implementada usa os apontadores: pai, flo, nível e arco que liga ao pai.

CAPÍTULO 2

ESTRUTURA INTERNA DO SISTEMA ESPECIALISTA RD

Este capítulo descreve detalhadamente a estrutura interna do sistema especialista RD, que compreende os módulos Base de Conhecimento, Máquina de Inferência e Base de Dados Geral. Do ponto de vista da arquitetura geral do sistema, estes módulos compõem a parte de programação simbólica, implementados na linguagem PROLOG. Os outros módulos, Gerenciador da Rede e Simulador de Manobra, são considerados como módulos de apoio e estão detalhados no capítulo 3.

Na seção 2.1. é descrita a representação do conhecimento adotada para o RD: fatos na forma de triplas Objeto-Atributo-Valor e regras de produção.

A estrutura básica de consulta utilizada no RD, árvore de contexto, é apresentada na seção 2.2., onde explana-se a forma de armazenamento do conhecimento sobre as relações existentes entre os objetos tratados pelas regras. Um exemplo de raciocínio utilizando a estrutura genérica de árvore de contexto é apresentado na seção 2.3.

A árvore de contexto particular para a resolução do problema de monobras em redes de distribuição é descrita na seção 2.4.

Na seção 2.5. são definidas as funções para

avaliação das premissas e as funções para execução das ações.

Uma relação das regras de produção na forma traduzida é fornecida nesta seção 2.6.

A seção 2.7. trata do mecanismo de inferência implementado no sistema RD. Os módulos que compõem a máquina de inferência estão descritos e apresentados na forma de diagramas de ação.

Finalmente, a seção 2.8. ilustra, com um exemplo comentado, uma consulta ao RD.

2.1 REPRESENTAÇÃO DO CONHECIMENTO

O conhecimento de um especialista pode ser representado de diferentes maneiras, entre as quais podemos destacar: Redes Semânticas, triplas O-A-V (Objeto-Atributo-Valor), Regras, Frames, Esquema e expressões lógicas [WIN1, RIC1, HAR1]. Todas elas, de alguma forma, devem representar objetos e descrever as relações existentes entre eles.

No sistema RD o conhecimento é representado por Fatos e Regras. Os fatos são representados por triplas do tipo O-A-V. Nesta representação, o objeto pode representar entidades físicas como uma *chave de manobra* ou ainda entidades conceituais como um *alimentador*. Atributos são características ou propriedades associadas aos objetos.

Código e *tipo* são atributos do objeto chave. Por último, o valor estabelece a natureza particular de um atributo em uma dada situação ou instante. O código (um atributo) da chave pode ter, por exemplo, o valor 120 ou a chave é do tipo (outro atributo) que permite sua abertura com carga.

A figura 2.1 mostra o exemplo de um fato na forma O-A-V. Neste caso o objeto é bloco com defeito e o atributo seu código. No exemplo em particular o *código* do *bloco com defeito* é 120 (ou seja, o valor do atributo código é 120).

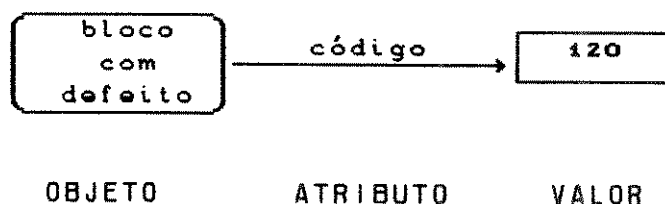


Figura 2.1. Fato na forma O-A-V.

Na linguagem PROLOG o exemplo da figura 2.1. é implementado pela *cláusula*:

```
fato(blcdef,cod,int(120)).
```

Nesta cláusula PROLOG, *blcdef* é o objeto bloco com defeito, *cod* é o atributo código e *int(120)* seu valor neste instante (uma instância para o atributo código).

É importante distinguir entre o *fato estático*, com seus atributos genéricos e uma *instância dinâmica* deste

fato, que se altera caso a caso. Se for retirada a parte da tripla O-A-V referente a um caso particular (denotado valor), obtém-se a representação estática do conhecimento, exemplificado na figura 2.2.

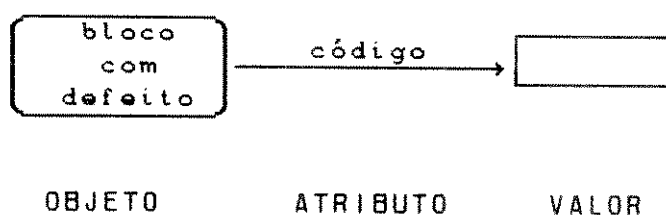


Figura 2.2. Representação estática da tripla O-A-V.

Pode-se agora diferenciar precisamente entre o conteúdo da Base de Conhecimento e o da Base de Dados Gerais ("Quadro Negro"). Na base de conhecimento está armazenado o conhecimento estático (fatos estáticos e as regras). Durante uma consulta ao RD o sistema busca determinar os valores dos atributos relacionados com uma determinada situação, que são armazenados na base de dados gerais como uma instância dinâmica dos fatos. Este processo é comumente denominado de *instanciamento*¹.

As regras são usadas para descrever o relacionamento entre os fatos, possibilitando à máquina de

¹ Derivado do verbo em inglês "to instance" que, para efeito de compreensão, adotaremos como tradução o neologismo "instanciar" que significa tornar uma abstração concreta em um dado instante.

inferência do RD, a partir dos fatos existentes em um dado instante, inferir novos fatos que conduzam à solução do problema.

O sistema RD utiliza um tipo de modelamento do conhecimento sobre manobras em rede de distribuição denominado na literatura [RIC1] regras de produção. As regras são do tipo SE *premissa* ENTÃO *ação*. A premissa é uma conjunção de condições (cláusulas) ligadas pelo operador lógico E. Da mesma forma, a ação é a conjunção de conclusões ligadas pelo operador lógico E. A figura 2.3. ilustra uma regra típica do sistema RD.

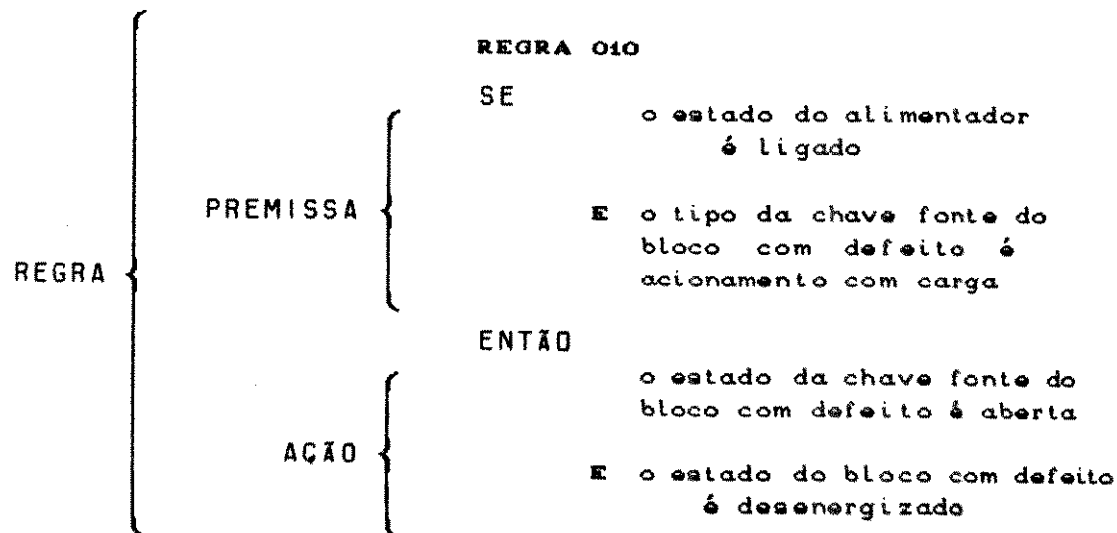


Figura 2.3. Regra típica do sistema RD.

Tanto as condições como as conclusões são expressas por triplas O-A-V, como pode ser notado na figura 2.4. (a ordem O-A-V está parcialmente invertida na figura

para efeito de compreensão).

	ATRIBUTO	OBJETO	VALOR
SE	Estado	Alimentador	Ligado
	Tipo chave fonte	Bloco com def	Acion. com carga
ENTÃO	Est. chave fonte	Bloco com def	Aberta
	Estado	Bloco com def	Desenergizado

Figura 2.4. Regras como triplas O-A-V.

A sintaxe das regras na linguagem PROLOG é apresentada a seguir na representação denominada Bachus Nauer Form (BNF):

<regra> ::= regra(<número da regra>,<premissa>,<ação>).

<premissa> ::= premissa(<se-cláusula>)

<se-cláusula> ::= <condição> | e(<se-cláusula>)

<condição> ::= cond(<função-p>,<atributo>,<contexto>,
<valor>)

<ação> ::= ação(<então-cláusula>)

<então-cláusula> ::= <conclusão> | e(<então-cláusula>)

<conclusão> ::= conc(<função-a>,<atributo>,<contexto>,
<valor>)

<número da regra> ::= <inteiro>

<função-p> ::= igual | maior | existe | busca | todos

<função-a> ::= conclua | mostre | operar

<contexto> ::= rede | blocdef | alimentador | manobra | chave

<valor> ::= sy(<símbolo>) | i(<inteiro>) | nulo

```

<atributo> ::= <símbolo>
<inteiro> ::= <dígito> | <dígito> <inteiro>
<símbolo> ::= <letra> | <letra> <símbolo>
<dígito> ::= 0 | 1 | ... | 9
<letra> ::= a | b | ... | x | y | z

```

Por exemplo, a regra da figura 2.3. fica implementada na linguagem PROLOG da seguinte forma:

```

regra(10,premissa(e(cond(igual,est_alim,alimentador,
                                sy(lig)),
                        cond(igual,tipo_chvf,bicdef,
                                sy(com)))),
ação      (e(conc(operar,chvf,bicdef,sy(abrir)),
              conc(conclua,est_bld,bicdef,
                    sy(desl)))))).

```

Esta regra representa a parte do conhecimento para julgar se o bloco com defeito foi ou não desligado.

2.2. ESTRUTURA BÁSICA DE CONSULTA

Conforme foi apresentado na seção 2.1., os fatos são representados por triplas O-A-V. Os objetos são as entidades físicas ou lógicas conhecidas como *contexto* no qual são aplicadas as regras. Os tipos de contexto são relacionados, hierarquicamente, pela árvore de contexto.

Árvore de contexto é a estrutura básica da consulta ao sistema RD, organizando a estrutura conceitual da base de conhecimento, definindo o fluxo de controle durante a consulta e indicando as metas que devem ser

derivadas pela máquina de inferência. Cada nó da árvore de contexto representa um tipo de contexto reconhecido pelo sistema RD. Os tipos de contexto descrevem a forma de instanciar o contexto. Existem dois aspectos relacionados na árvore de contexto: a árvore estática dos tipos de contexto e a árvore dinâmica das instâncias dos contextos. Esta distinção faz-se importante durante o processo de raciocínio da máquina de inferência, porque é a árvore estática que orienta a criação da estrutura em árvore dos instanciamentos dos tipos de contexto. Esse processo cria uma árvore dinâmica a cada consulta ao sistema RD.

A figura 2.5. esquematiza o instanciamento de uma árvore de contexto genérica, contendo 4 tipos de contexto relacionados hierarquicamente.

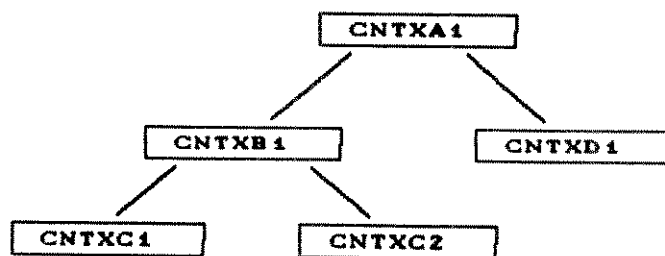


Figura 2.5. Exemplificação da árvore de contexto

Para cada árvore de contexto deve existir um e somente um tipo de contexto, denominado *raiz*, que será instanciado uma única vez durante a consulta. O contexto CNTXA1 é a raiz da árvore de contexto da figura 2.5. Os demais tipos de contexto podem ser instanciados mais de uma

vez. Na figura 2.5. existem duas instâncias para o tipo de contexto C: CNTXC1 e CNTXC2.

Durante o processo de raciocínio da máquina de inferência os nós da árvore de contexto são criados dinamicamente. O mecanismo de criação destes nós utiliza uma estrutura de dados para derivar o instanciamento dos tipos de contexto. Cada tipo de contexto é caracterizado pelos seguintes parâmetros:

Tcntx	:	nome interno para o tipo de contexto.
Dados Iniciais	:	lista dos atributos que devem ser instanciados quando da criação do contexto.
Objetivo	:	meta para máquina de inferência.

A estrutura de dados para os tipos de contexto são implementadas como cláusulas PROLOG. Por exemplo:

```
tipo_cntx(rede,"REDE",[],[plano_rest]).
```

Esta cláusula implementa o tipo de contexto *rede*.

2.3. RACIOCÍNIO POR CONTEXTO

Cada nó da árvore estática tem um ou mais atributos que devem ser instanciados para defini-lo na árvore dinâmica. O objetivo da máquina de inferência é derivar as instâncias de cada um dos atributos da árvore de contexto. Na figura 2.6. CONTEXT01 é o contexto raiz e o atributo objetivo é ATRIB1.

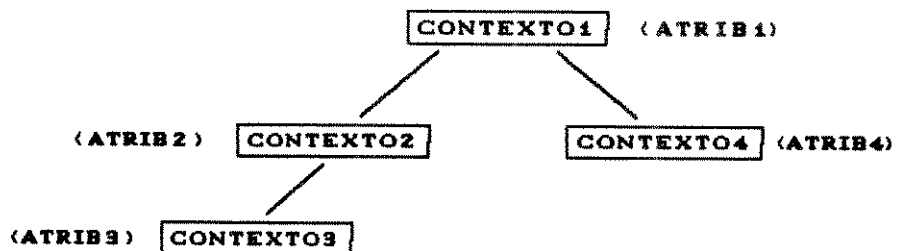


Figura 2.6.Árvore de contexto

Inicialmente a máquina de inferência verifica se já foi determinado o ATRIB1 do CONTEXT01. Se sim, termina o raciocínio. Caso contrário, obtém o conjunto de regras cuja ação leva à conclusão do ATRIB1 do CONTEXT01 e inicia a avaliação da premissa da regra selecionada. Quando a premissa de uma regra faz referência a contextos ainda não criados, um nó na árvore dinâmica deve ser criado e portanto o tipo de contexto instanciado.

Suponhamos que o CONTEXT02 seja instanciado pelo mecanismo de criação. Neste instante a máquina de inferência

procura derivar a instância do ATRIB2, para tanto, torna-o um sub-objetivo a ser perseguido e reinicia o raciocínio, obtendo o conjunto de regras para inferir o ATRIB2 do CONTEXT02. Uma vez que todos os atributos dos contextos abaixo da raiz da árvore de contexto tenham sido exemplificados, o atributo objetivo fica determinado.

Nota-se que este processo de instanciamento dos atributos e a determinação, hierarquizada, dos nós da árvore dinâmica definem o fluxo de controle da máquina de inferência. Quando a tentativa de instanciamento de um atributo de um contexto filho falha, a máquina de inferência realiza o "backtracking" ao nó pai na busca de outras alternativas de solução do problema.

2.4. ÁRVORE DE CONTEXTO DO SISTEMA RD

Como foi visto na seção 2.3., uma árvore de contexto é constituída de nós dos tipos de contexto e cada um deles tem um ou mais atributos a serem instanciados para definir o nó na árvore dinâmica. A figura 2.7. mostra a árvore de contexto usada no sistema RD.

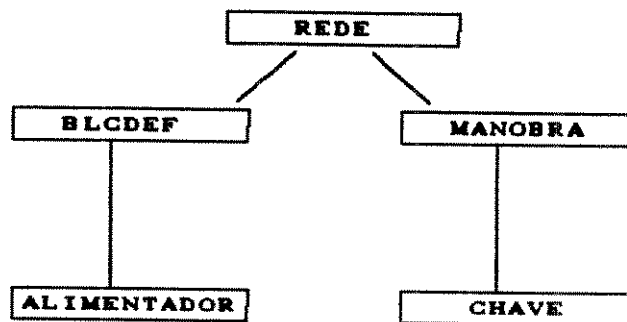


Figura 2.7. Árvore de contexto do sistema RD.

Os tipos de contexto reconhecidos atualmente pelo RD são:

REDE	:	A rede de distribuição primária (13.8 kV).
BLCDEF	:	Bloco de carga onde ocorreu uma contingência.
ALIMENTADOR	:	Alimentador do bloco de carga com problema.
MANOBRA	:	Conjunto de operações para o restabelecimento da rede.
CHAVE	:	Chave candidata a restabelecer a rede.

Os tipos de contexto instanciados durante uma consulta ao RD são dispostos hierarquicamente na estrutura de dados denominada árvore dinâmica. A figura 2.8. ilustra uma árvore dinâmica, que pode ser gerada numa consulta onde

existem 3 chaves candidatas a restabelecer a rede de distribuição.

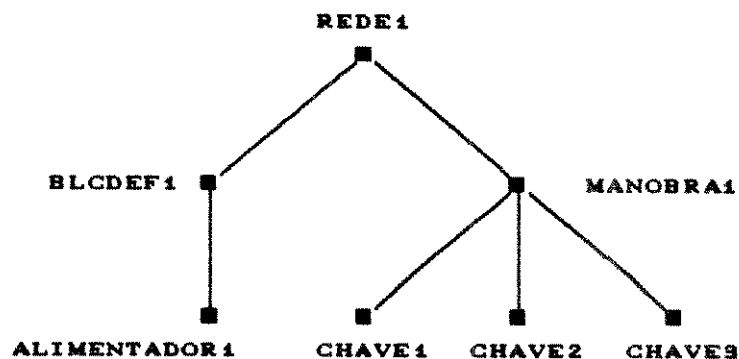


Figura 2.8. Exemplo de uma árvore dinâmica.

Nota-se na figura 2.8. que um tipo de contexto pode ser instanciado mais de uma vez, isto é, gerar mais de um contexto. Como neste exemplo são 3 chaves candidatas à manobra de restabelecimento, foram criados os contextos CHAVE1, CHAVE2 e CHAVE3. A criação destes contextos faz parte do processo de enumeração das candidatas à manobra. Quando um destes contextos satisfaz as condições pré-estabelecidas de queda de tensão e capacidade física, o processo termina realizando o "backtracking" até o nó raiz, mostrando a solução do problema.

Como foi visto anteriormente, a cada contexto é associado um conjunto de propriedades ou características. Este conjunto de propriedades compreende os atributos que devem ser instanciados para definir um nó da árvore de

contexto. Os atributos associados a cada contexto do sistema RD são:

CONTEXTO	ATRIBUTO	DESCRIÇÃO
REDE	plano_rest	plano de restabelecimento
	restab	se a rede foi ou não restabelecida.
BLCDEF	cod	código do bloco com defeito.
	tipo	tipo do bloco com defeito (fim de linha ou fonte/carga).
	chvf	Código da chave fonte do bloco com defeito.
	chvc	Código da chave carga do bloco com defeito.
	tipo_chvf	tipo da chave fonte (acionamento com carga ou não).
	est_bld	estado do bloco com defeito (energizado ou deenergizado).
	interr	se a interrupção foi ou não isolada.

(cont.)

CONTEXTO	ATRIBUTO	DESCRIÇÃO
ALIMENTADOR	est_alim	estado do alimentador (ligado ou desligado).
	chvtipo	tipo da chave no alimentador (acionamento com carga ou não).
	chv	chave de manobra para isolar o defeito.
	dsj	código do disjuntor do alimentador.
MANOBRA	num_cand	número de chaves candidatas para manobra.
	candid	código da chave candidata a manobra.
	chv_mnbr	código da chave da manobra de restabelecimento.
CHAVE	marca	se a chave foi ou não enumerada.
	est_bicf	estado (energizado ou desenergizado) do bloco fonte da chave de manobra.
	simulação	resultado da simulação da manobra de restabelecimento.

(cont.)

CONTEXTO	ATRIBUTO	DESCRIÇÃO
CHAVE	est_alim_bicf	estado (energizado ou desenergizado) do alimentador do bloco fonte da chave de manobra.

Estes atributos são armazenados em uma estrutura de dados com os seguintes parâmetros:

Nome	:	símbolo interno ao sistema.
Tipo	:	este parâmetro indica os valores esperados para o atributo, isto é, lstsy - lista de símbolos sy - símbolo i - inteiro r - real pred - tipo especial para indicar que a instância do atributo é obtida por uma cláusula PROLOG.
Alerta	:	Este parametro é usado com duas finalidades: a) Uma sentença utilizada para solicitar ao usuário o valor do atributo. b) o símbolo do predicado PROLOG usado para instanciar o atributo.
Regras	:	Este parâmetro contém a lista das regras que fazem referência ao atributo em sua ação.

(cont.)

Tradução : Este parâmetro é usado para traduzir a ocorrência do atributo em frases em Português.

O interfaceamento entre as linguagens PROLOG e C é conseguido pelos atributos do tipo *pred*. Os predicados em sua maioria têm o objetivo de obter os parâmetros necessários à função C desejada, fazer sua chamada e finalmente armazenar o resultado (quando existir) na base de dados gerais.

2.5. FUNÇÕES PARA AVALIAÇÃO DAS REGRAS

As funções para avaliação das regras estão classificadas em *função-p* e *função-a*, aplicadas, respectivamente, às premissas e às ações das regras de produção do sistema RD.

2.5.1. Funções de Avaliação das Premissa

A avaliação das condições na premissa das regras resulta em VERDADEIRA ou FALSA. A máquina de inferência para chegar a esta conclusão (condição verdadeira ou falsa) utiliza-se de predicados declarados nas regras como *função-p*. A seguir são listadas as funções-p implementadas

nesta versão do RD

FUNÇÃO-P	SE	ENTÃO	SENÃO
<i>igual</i> (VALOR1,VALOR2)	VALOR1 = VALOR2	V	F
<i>maior</i> (VALOR1,VALOR2)	VALOR1 > VALOR2	V	F
<i>existe</i> (Atrib,Cntx)	existe o Atrib. do Contexto na Base de Dados Geral.	V	F
<i>busca</i> (Atrib,Cntx)	é possível instanciar o Atrib. do Contexto.	V	F

Além dessas existe a função-p *todos*(Lista-Candidatas), que implementa o processo de enumeração dos elementos da lista de candidatas à monobra. Esta função-p permite enumerar as chaves candidatas à manobra na busca de uma solução do problema.

2.5.2 Funções de Execução das Ações

Se a avaliação da premissa de uma regra resultar em verdadeira, a máquina de inferência executa uma ação. Para tanto, utiliza-se de funções do tipo *função-a* para realizar as conclusões pertinentes. A seguir estão relacionadas as funções-a implementadas no sistema RD:

FUNÇÃO-A	DESCRIÇÃO
<i>conclua(Atrib,Cntx,Valor)</i>	esta função armazena a tripla C-A-V (ou seja O-A-V) na Base de Dados Geral.
<i>operar(Atrib,Cntx,Ação)</i>	esta função realiza a abertura ou fechamento de chaves da rede de distribuição. O parâmetro Ação está definido na regra.
<i>mostre(Atrib,Cntx)</i>	função para mostrar os fatos armazenados na Base de Dados Geral para o Atributo do Contexto.

A função-a *operar* age sobre a estrutura de dados da rede e, portanto, utiliza-se de funções na linguagem C para atualizar os dados.

2.6. REGRAS DO SISTEMA RD

A base de conhecimento do sistema RD conta, atualmente, com 13 regras. Estas regras, obtidas a partir de normas técnicas [CPFL1, CPFL3, CPFL4] e manual de treinamento [MAT1], estão listadas a seguir na forma traduzida. São regras básicas que representam o conhecimento equivalente a um operador com pouca experiência prática.

REGRA 001:

SE o bloco com defeito foi isolado E
 a rede de distribuição foi restabelecida

ENTÃO mostre o plano de manobra

REGRA 005:

SE o alimentador está desligado

ENTÃO abrir a chave fonte do bloco com defeito E
 fechar o disjuntor do alimentador E
 conclua bloco com defeito desligado

REGRA 010:

SE o alimentador está ligado E
 a chave fonte do bloco com defeito pode ser
 acionada com carga

ENTÃO abrir a chave fonte do bloco com defeito E
 conclua bloco com defeito desligado

REGRA 015:

SE o alimentador está ligado E
 a chave fonte do bloco com defeito não pode ser
 acionada com carga E
 existe chave no alimentador que pode ser acionada
 com carga

ENTÃO abrir chave no alimentador E
 abrir chave fonte do bloco com defeito E
 fechar chave no alimentador E
 conclua bloco com defeito desligado

REGRA 017:

SE o bloco com defeito está desligado E
 o bloco é fim de linha

ENTÃO conclua bloco com defeito isolado

REGRA 020:

SE *bloco com defeito está desligado E*
 bloco com defeito é fonte/carga E
 é possível encontrar chave carga do bloco com
 defeito

ENTÃO *abrir chave carga do bloco com defeito E*
 conclua bloco com defeito isolado

REGRA 025:

SE *o bloco com defeito é fim de linha E*
 o bloco com defeito foi isolado

ENTÃO *rede de distribuição restabelecida*

REGRA 030:

SE *não existem chaves candidatas para manobra*

ENTÃO *conclua rede não pode ser restabelecida*

REGRA 035:

SE *existem chaves candidatas para manobra E*
 existe chave para manobra de restabelecimento

ENTÃO *fechar chave para restabelecer a rede E*
 conclua rede restabelecida

REGRA 040:

SE *o bloco fonte da chave candidata está energizado E*
 simulação da manobra no limite

ENTÃO *conclua chave candidata pode ser manobrada*

REGRA 045:

SE *o bloco fonte da chave candidata está energizado E*
 simulação da manobra fora do limite

ENTÃO *conclua chave candidata não pode ser manobrada*

REGRA 050:

SE *o alimentador do bloco fonte da chave candidata
 está desenergizado*

ENTÃO *conclua chave candidata não pode ser manobrada*

REGRA 055:

SE *é possível encontrar a chave de acionamento com
 carga no alimentador*

ENTÃO *conclua chave de acionamento com carga do
 alimentador encontrada*

2.7. MECANISMO DE INFERÊNCIA

O módulo de maior importância em um sistema especialista é a Máquina de Inferência. Ela encerra todo mecanismo de raciocínio necessário à condução de uma consulta feita ao sistema. A máquina de inferência tem dois objetivos principais. Primeiro, avaliar fatos e regras existentes e, quando possível, acrescentar novos fatos à base de conhecimento. Segundo, decidir a ordem em que as regras são avaliadas. Existem várias maneiras de implementar esses objetivos. No sistema RD foi implementada uma máquina de inferência baseada nos conceitos usados no sistema especialista MYCIN [BUC1].

A estratégia de inferência usada no RD é do tipo "SE a premissa é verdadeira ENTÃO execute a ação" (Modus Ponens). O controle de inferência é dirigido por metas ("backward chaining"). O raciocínio suportado é monotônico,

isto é, todos os fatos inferidos permanecem até o fim de uma consulta ao sistema [HAR1, RIC1].

O controle de inferência dirigido por metas é implementado por dois módulos interrelacionados: *AVALIADOR* e *PROCURA* (Figura 2.9.).

O *AVALIADOR* analisa cada condição da premissa da regra e se todas as condições forem verdadeiras ele executa a ação. Caso contrário, a regra é rejeitada. Durante a análise da premissa o *AVALIADOR* verifica se todos os dados necessários à avaliação da regra já se encontram na base de conhecimento. Se um contexto não existe ou um atributo ainda não se encontra na base de conhecimento o avaliador aciona o mecanismo de busca do *PROCURA*.

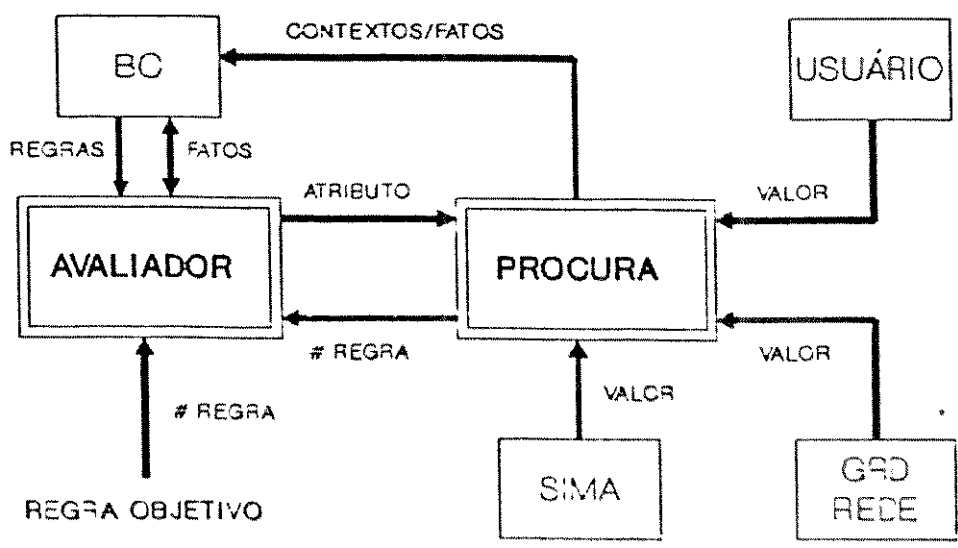


Figura 2.9. Máquina de Inferência do RD.

O módulo PROCURA cria um contexto ou obtém um valor que instancie o atributo solicitado. O instanciamento de um atributo pode levar este módulo a solicitar o valor desconhecido ao usuário, acionar o módulo Gerenciador da Rede ou acionar o módulo Simulador de Manobra para obter dados sobre o estado (físico e elétrico) e interrelacionamento dos elementos da rede, ou ainda, inferir o valor por meio de regras a serem analisadas pelo AVALIADOR.

O processo AVALIADOR $\rightleftarrows$ PROCURA é recursivo, permite um raciocínio em cadeia, criando uma árvore de contexto dinamicamente, até a solução do problema. Para disparar este processo é necessário uma regra denominada *regra objetivo*. A regra objetivo traduz os problemas a serem resolvidos pela máquina de inferência, assim como, as ações a serem executadas quando uma solução é encontrada.

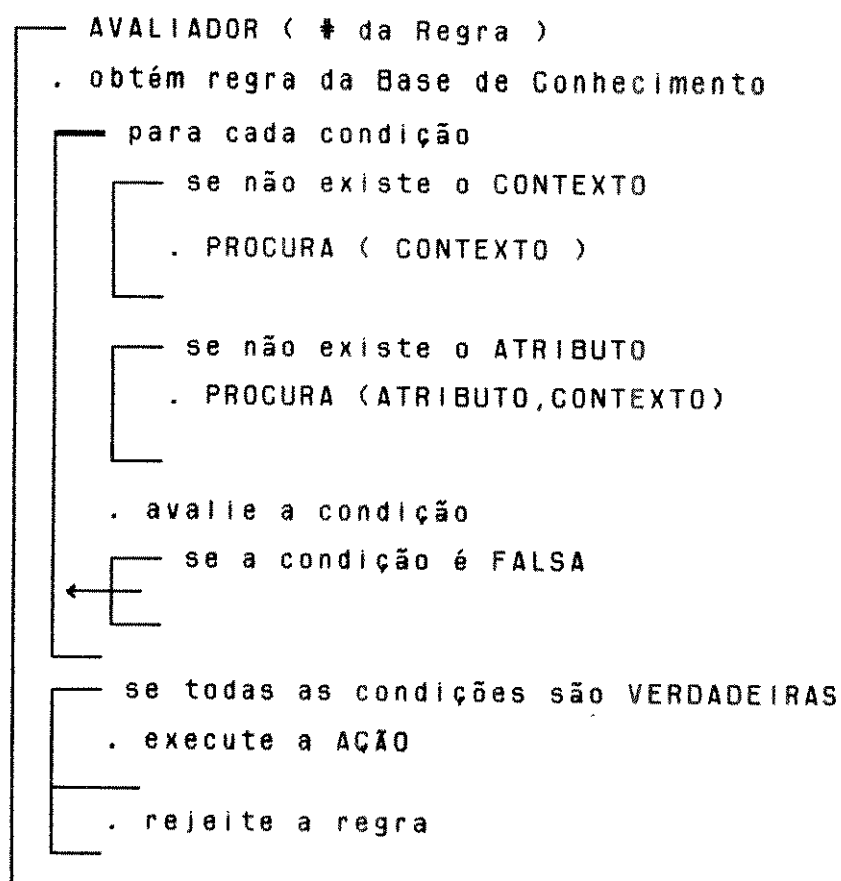
No início de cada consulta ao RD, a máquina de inferência cria o contexto *rede1* e busca na base de conhecimento a regra objetivo para análise do AVALIADOR. A regra objetivo do RD deixa explícitos as tarefas a serem executadas pelo sistema:

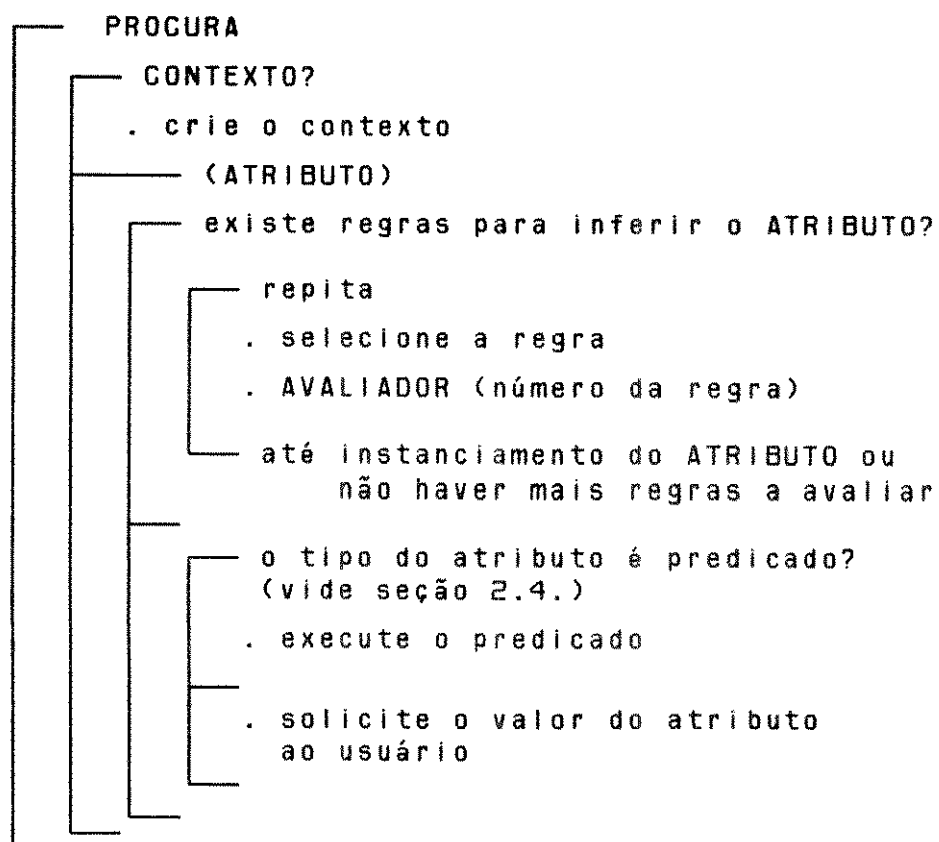
SE *o bloco com defeito foi isolado E*
 a rede foi restabelecida
ENTÃO *apresente o plano de manobras.*

As tarefas explicitadas pela regra objetivo acima

são: primeiro, isolar da rede em operação o bloco com defeito; depois, procurar uma solução de restabelecimento da área escura.

A seguir estão detalhados os módulos AVALIADOR e PROCURA na forma de diagrama de ação.





2.8. EXEMPLO DE UMA CONSULTA AO SISTEMA RD

O exemplo descrito a seguir ilustra uma consulta ao sistema RD, aplicado à rede de distribuição primária da figura 2.10.

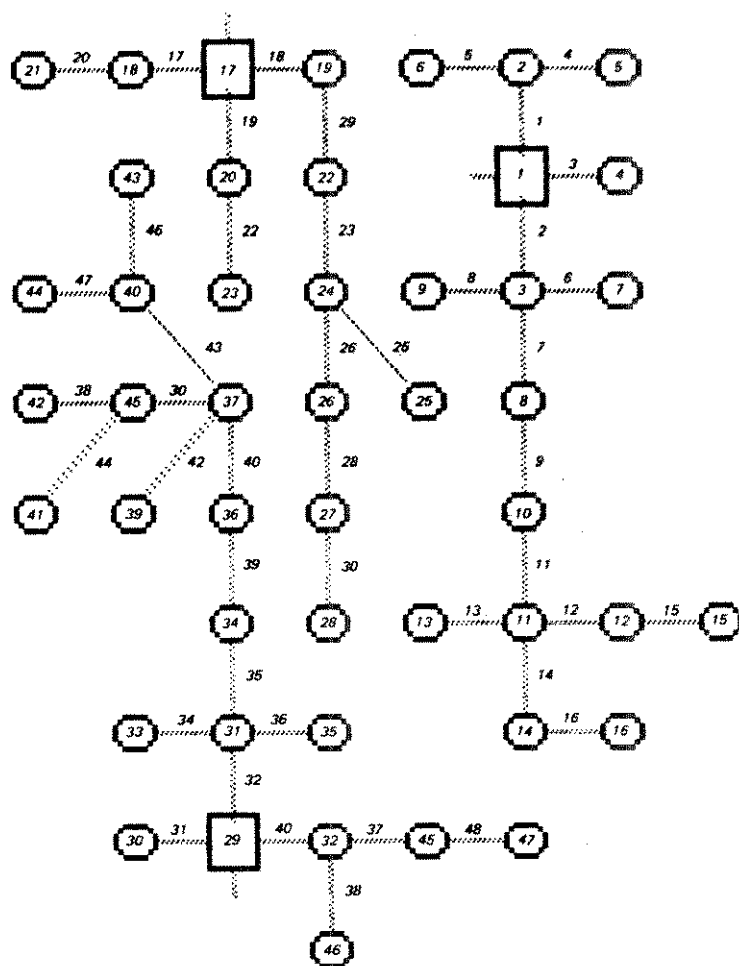


Figura 2.10. Rede de distribuição primária.

Após a leitura dos dados da rede, o RD cria o contexto rede1 e inicia a avaliação da REGRA 001 (regra objetivo). Na avaliação da primeira condição desta regra, o RD cria o contexto bicdef1 para buscar as manobras de isolamento do bloco com defeito. Neste instante é necessário conhecer o código do bloco com defeito. Porém como esta informação não se encontra na Base de Conhecimento, então, o RD solicita ao usuário, conforme exemplificado a seguir:

RD versão 1.0

Lendo Dimensão do Problema	}	<i>leitura dos dados da rede</i>
Lendo Dados de Nó		
Lendo Dados de Arcos		
Fim da Leitura		
--- rede1 ---	}	<i>cria contextos</i>
--- bicdef1 ---		
Qual o código do bloco com defeito? _		

Seja o bloco 36 (na figura 2.10.) o código do bloco com defeito. Com esta informação o RD pode inferir outros atributos do contexto bicdef1 como tipo do bloco, chave fonte e seu tipo; e mostra ao usuário os fatos que estão sendo armazenados na Base de Conhecimento:

RD versão 1.0

Lendo Dimensão do Problema

Lendo Dados de Nó

Lendo Dados de Arcos

Fim da Leitura

--- rede1 ---

--- bldcdef1 ---

Qual o código do bloco com defeito? 36

FATO: bldcdef1,cod_bcd,36 [*código do bloco*
 com defeito 36]

[*atributos inferidos do bloco com defeito*]

FATO: bldcdef1,tipo,fc [*o bloco e do tipo*
 fonte/carga]

FATO: bldcdef1,chvf,39 [*a chave fonte tem*
 o código 39]

FATO: bldcdef1,tipo_chvf,sem [*a chave fonte*
 só pode ser
 acionada sem
 carga]

Para inferir se o bloco com defeito foi interrompido ou não, existem duas regras na base de conhecimento: REGRA D17 e REGRA D20. O RD escolhe a REGRA D17 para avaliar, a qual necessita saber o *estado do bloco* (energizado ou desenergizado). A base de conhecimento neste instante não contém nenhum fato sobre o estado do bloco, então o RD seleciona REGRA D05, REGRA D10 e REGRA D15 para inferir o atributo desejado.

O estado do bloco depende a princípio do estado do alimentador. O RD, então, cria o contexto alimentador1 e instancia seus atributos: código, disjuntor e seu estado

neste instante. A instância dos dois primeiros atributos são obtidos com auxílio do módulo GERENCIADOR DA REDE e informa ao usuário:

[*atributos referentes ao alimentador*]

FATO: alimentador1,cod,29 [*codigo 29*]

FATO: alimentador,dsj,32 [*codigo do disjuntor*]

O estado do alimentador em conjunto com o tipo da chave fonte do bloco com defeito definem o tipo de manobra a ser executada. O RD solicita ao usuário o estado atual do alimentador:

```
--- rede1 ---
--- b1cdef1 ---
Qual o código do bloco com defeito? 36
FATO: b1cdef1,cod_bcd,36
FATO: b1cdef1,tipo,fc
FATO: b1cdef1,chvf,39
FATO: b1cdef1,tipo_chvf,sem
--- alimentador1 ---
FATO: alimentador1,cod,29
FATO: alimentador,dsj,32
Qual o estado atual do alimentador? _
```

Suponhamos que o alimentador esteja ligado, isto é, o sistema de proteção do alimentador não foi acionado. O usuário informa esta condição digitando lig e o RD acrescenta este fato a base de conhecimento:

```

--- rede1 ---
--- b1cdef1 ---
Qual o código do bloco com defeito? 36
FATO: b1cdef1,cod_bcd,36
FATO: b1cdef1,tip0,fc
FATO: b1cdef1,chvf,39
FATO: b1cdef1,tip0_chvf,sem
--- alimentador1 ---
FATO: alimentador1,cod,29
FATO: alimentador,dsj,32
Qual o estado atual do alimentador? 11g
FATO: alimentador1,est,11g [ o alimenta-
                             dor está
                             11gado ]

```

A REGRA005 é rejeitada porque é aplicada na condição do alimentador estar desligado. O RD passa a avaliar a REGRA 010, que também é rejeitada porque se aplica a chaves que podem ser acionadas com carga. A REGRA 015 é escolhida e o objetivo agora é encontrar no alimentador1 uma chave que possa ser aberta com carga. Esta resposta é obtida pela avaliação da REGRA 055. O RD indica os novos fatos inferidos:

```

FATO: alimentador1,chv,32 [ chave de código 32 ]
FATO: alimentador1,chvtipo,com [ pode ser acio-
                                nada com carga ]

```

Com estes fatos armazenados na base de conhecimento, o RD aceita a premissa da REGRA 015 como verdadeira e, conseqüentemente, conclui as manobras de desligar o bloco com defeito, assim como seu estado

(desligado):

```
--- rede1 ---
--- blcdef1 ---
Qual o código do bloco com defeito? 36
FATO: blcdef1,cod_bcd,36
FATO: blcdef1,tipofc
FATO: blcdef1,chvf,39
FATO: blcdef1,tipochvf,sem
--- alimentador1 ---
FATO: alimentador1,cod,29
FATO: alimentador,dsj,32
Qual o estado atual do alimentador? lig
FATO: alimentador1,est,lig
FATO: alimentador1,chv,32
FATO: alimentador1,chvtipo,com

[ Plano de Manobra de isolamento ]

FATO: plan_rest,32,abrir [ abrir chave 32 ]
FATO: paln_rest,39,abrir [ abrir chave 39 ]
FATO: plan_rest,32,fechar [ fechar chave 32 ]
FATO: blcdef1,est,desl [ bloco com defei-
                        to foi desligado ]
```

Observe que este tipo de manobra é conhecido como "pick" no alimentdor, isto é, foi necessário desligar o alimentador antes de operar a chave fonte do bloco com defeito. A rede neste instante está ilustrada na figura 2.11.

Sendo o bloco com defeito do tipo fonte/carga, há a necessidade de ser aberta a chave carga do bloco com defeito para que a manobra de isolamento seja completa. Isto faz com que o RD rejeite a REGRA D17 porque esta é aplicada para blocos fim de linha.

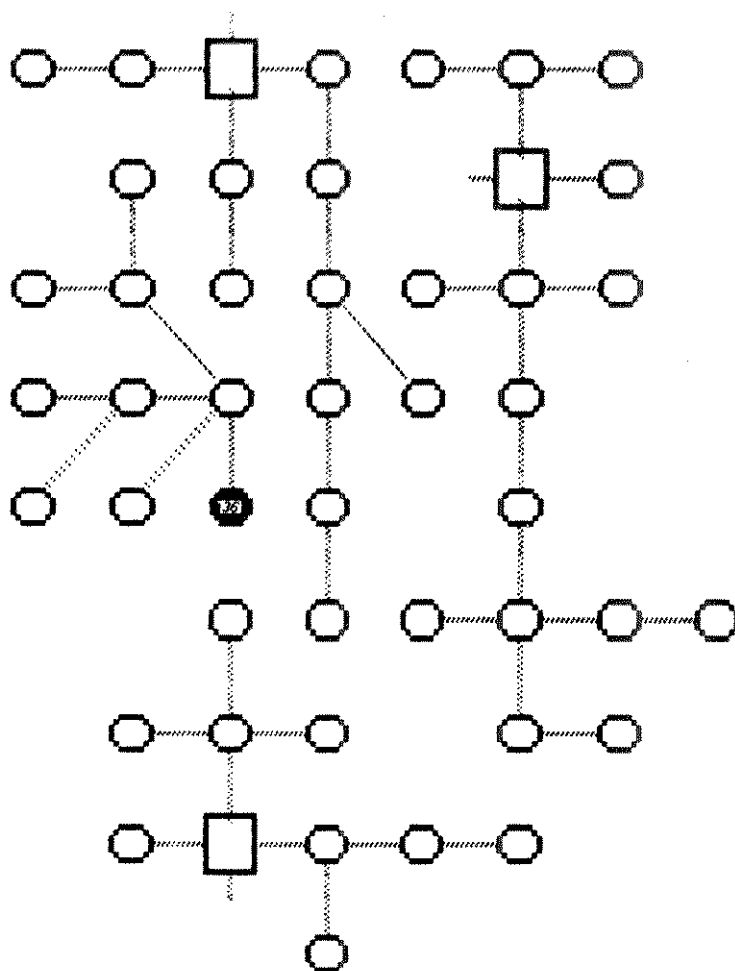


Figura 2.11. - Bloco com defeito desligado.

Continuando o raciocínio, o RD escolhe a REGRA 020 para ser avaliada, que tem as duas primeiras condições da premissa verdadeiras com relação aos fatos existentes na base de conhecimento. A avaliação da terceira condição conduz o RD a inferir a chave carga do bloco com defeito:

FATO: blcdef1,chvc,40 [*codigo da chave carga 40*]

A premissa da REGRA 020 sendo verdadeira o RD
conclui a manobra de isolamento do bloco com defeito:

```

--- rede1 ---
--- blddef1 ---
Qual o código do bloco com defeito? 36
FATO: blddef1,cod_bcd,36
FATO: blddef1,tipos,fc
FATO: blddef1,chvf,39
FATO: blddef1,tipos_chvf,sem
--- alimentador1 ---
FATO: alimentador1,cod,29
FATO: alimentador,dsj,32
Qual o estado atual do alimentador? lig
FATO: alimentador1,est,lig
FATO: plan_rest,32,abrir
FATO: plan_rest,39,abrir
FATO: plan_rest,32,fechar
FATO: blddef1,est,desl
FATO: blddef1,chvc,40

[ manobra que completa o Plano de manobra de
  Isolamento. ]
FATO: plan_rest,40,abrir [ abrir chave 40 ]

FATO: blddef,interrupção,isolada [ bloco com
                                   defeito
                                   isolado ]

```

A figura 2.12. mostra o bloco com defeito (nó mais
escuro) isolado da parte energizada da rede.

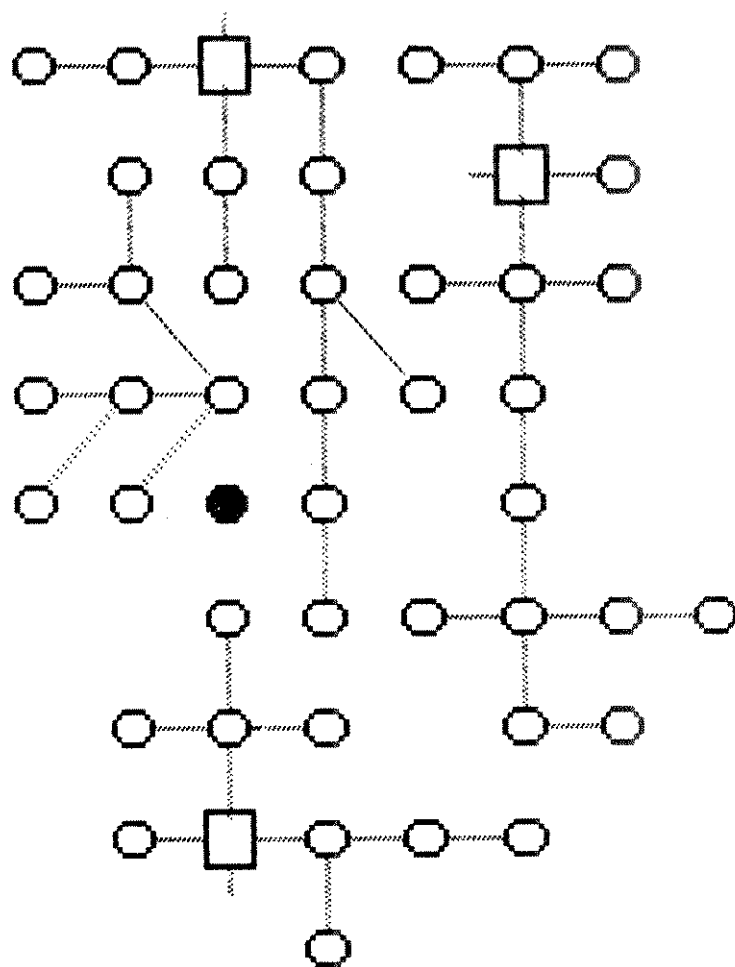


Figura 2.12. - Bloco com defeito isolado.

O objetivo do RD agora é a manobra a ser realizada para o restabelecimento da área escura (segunda premissa da REGRA 001). O RD seleciona as REGRA 025, REGRA 030 e REGRA 35 para serem avaliadas. A REGRA 025 é rejeitada porque aplica-se apenas a blocos com defeito fim de linha. A avaliação da REGRA 030 conduz a busca das chaves candidatas

ao restabelecimento da área escura. Para tanto, é criado o contexto *manobra1* e instanciado o número de candidatas com auxílio do módulo Gerenciador da Rede:

```

--- rede1 ---
--- blcdef1 ---
Qual o código do bloco com defeito? 36
:
fatos do blcdef1
:
--- alimentador1 ---
:
fatos do alimentador1
:
:
fatos do blcdef1
:
--- manobra1 ---
FATO: manobra1,num_cand,3 [ existem 3 chaves
                           candidatas a
                           manobra ]

```

Como existem 3 candidatas (chaves 40, 29 e 24 - figura 2.13) para manobra de restabelecimento a REGRA 030 é rejeitada. A próxima regra (REGRA 035) a ser selecionada coloca um novo objetivo ao RD: enumerar todas as chaves candidatas até encontrar uma que fechada não ultrapasse os limites pré-estabelecidos de queda de tensão e capacidade física.

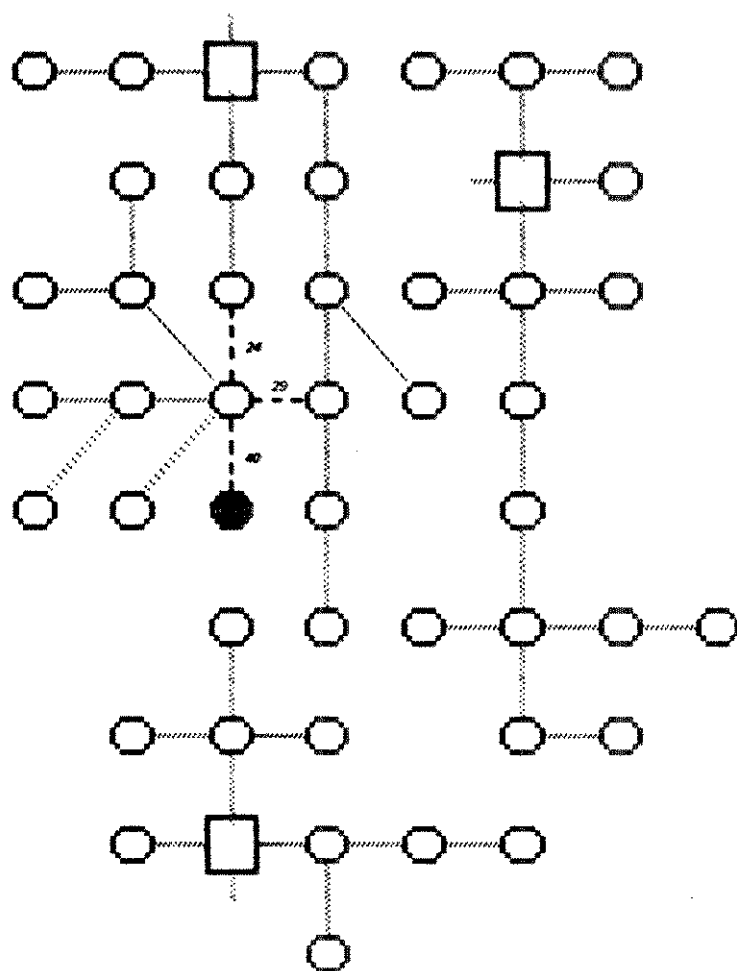


Figura 2.13. - Chaves candidatas

O RD cria um contexto para cada uma das candidatas e aplica a REGRA 040, REGRA 045 ou REGRA 050 para inferir a chave a ser manobrada. Inicialmente o RD solicita ao módulo Gerenciador da Rede a lista das candidatas, que neste exemplo são as chaves de código 40, 29 e 24 (figura 2.13.), enumeradas nesta ordem. Em seguida cria o contexto chave1 referente à chave 40. Para a avaliação da REGRA 040 é

necessário conhecer o estado do bloco fonte da chave1, o que é solicitado ao usuário:

```

--- rede1 ---
--- blcdef1 ---
Qual o código do bloco com defeito? 36
:
fatos do blcdef1
:
--- alimentador1 ---
:
fatos do alimentador1
:
:
fatos do blcde
:
--- manobra1 ---
FATO: manobra1,num_cand,3
--- chave1 ---
Qual é o estado do bloco fonte? _

```

Como o bloco fonte desta chave é o bloco com defeito e ainda não foi reparado, a resposta natural é desenergizado. O RD rejeita a REGRA D4D porque o bloco fonte da chave avaliada deve estar energizado. Como esta condição é imprescindível para restabelecer a área escura, o RD passa a avaliar a próxima chave e a mesma pergunta é colocada ao usuário, agora para a chave de código 29:

```

--- manobra1 ---
FATO: manobra1,num_cand,3
--- chave1 ---
Qual é o estado do bloco fonte? desenergizado_
--- chave2 ---
Qual é o estado do bloco fonte? _

```

O usuário informa o estado do bloco fonte da chave2, que neste caso é energizado. O RD aciona o módulo Simulador de Manobra para verificar se a manobra da chave2 (código 29) é aconselhável ou não. A figura 2.14. mostra a configuração da rede a ser simulada.

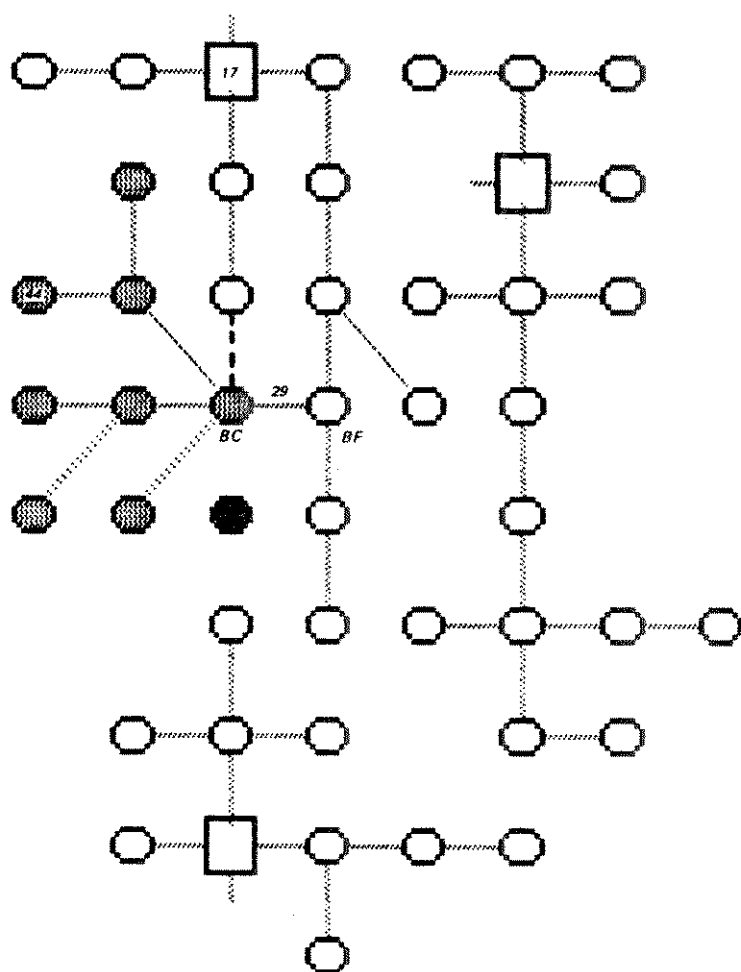


Figura 2.14 - Simulação da chave 29.

A seguir são apresentados os resultados obtidos da simulação.

1. Avalia a capacidade da Subestação suprir a carga.

Capacidade da SE[17]: 25.00 MVA

Carga a suprir : 11.88 MVA

2. Verifica a capacidade da chave.

Capacidade da chave 29: 400.00 A

Fluxo a suprir : 209.43 A

3. Verifica a queda de tensão no bloco fonte.

Limite de queda de tensão : 690.00 V

Queda de tensão no BF : 565.33 V

4. Verifica a queda de tensão no bloco carga.

Limite de queda de tensão : 690.00 V

Queda de tensão no BC : 650.89 V

5. Verifica máxima queda de tensão na AE

Limite de queda de tensão : 690.00 V

Máxima QT na área escura : 693.57 V no bloco 44

Manobra não aconselhável....

Note que a queda de tensão no bloco de carga de código 44 da área escura é 693,57V e, portanto, superior ao limite pré-estabelecido de 690,00V. Como o resultado da simulação está *fora do limite* a REGRA 040 é rejeitada.

A próxima chave candidata a ser avaliada é a de código 24, novamente é solicitado o auxílio ao usuário:

manobra1

FATO: manobra1,num_cand,3

chavel

Qual é o estado do bloco fonte? desenergizado.

0000- 0000- 0000- **chavez** - - - 0000

Qual é o estado do bloco fonte? energizado_

chave3

Qual é o estado do bloco fonte?

é energizado, nova simulação é solicitada para o módulo Simulador de Manobra para a configuração de rede ilustrada na figura 2.15.

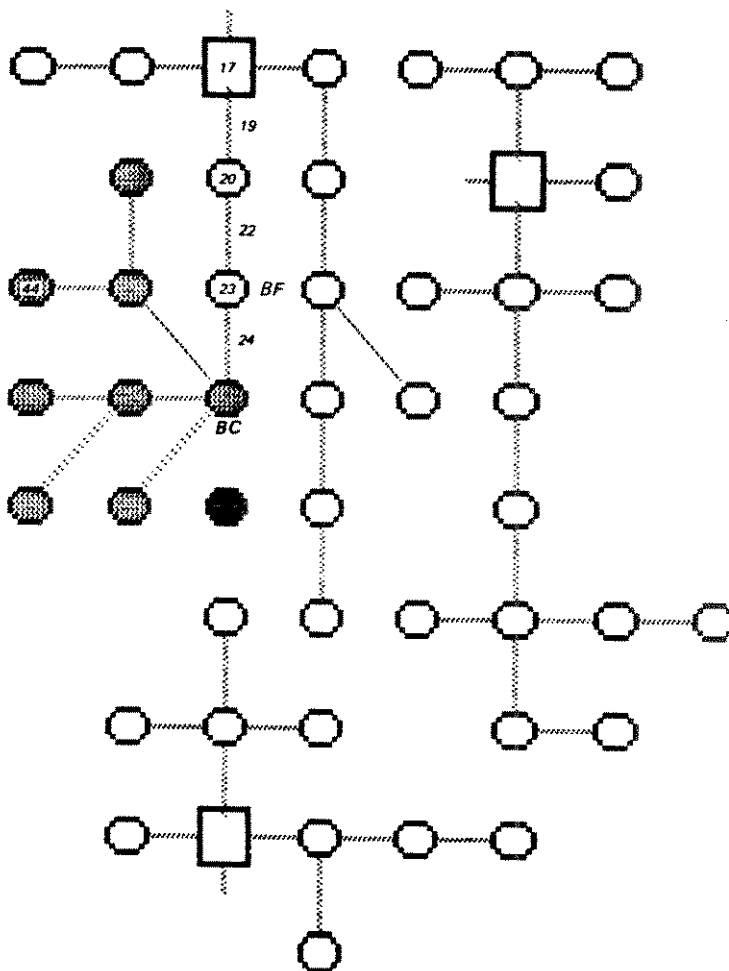


Figura 2.15. - Simulação da chave 24.

Os resultados da simulação da chave 24 são mostrados a seguir :

1. Avalia a capacidade da Subestação suprir a carga.

Capacidade da SE[17]: 25.00 MVA

Carga a suprir : 11.88 MVA

2. Verifica a capacidade da chave.

Capacidade da chave 24: 400.00 A

Fluxo a suprir : 209.43 A

3. Verifica a queda de tensão no bloco fonte.

Limite de queda de tensão : 690.00 V

Queda de tensão no BF : 202.74 V

4. Verifica a queda de tensão no bloco carga.

Limite de queda de tensão : 690.00 V

Queda de tensão no BC : 288.10 V

5. Verifica máxima queda de tensão na AE

Limite de queda de tensão : 690.00 V

Máxima QT na área escura : 330.78 V no bloco 44

6. Verifica a capacidade física do ALIMENTADOR.

Capacidade da chave 22: 400.00 A

Fluxo a suprir : 235.61 A

Capacidade da chave 19: 400.00 A

Fluxo a suprir : 261.79 A

7. Verifica máxima queda de tensão ALIMENTADOR.

Limite de queda de tensão : 690.00 V

Máxima QT no alimentador : 106.70 V no bloco 20

Note que nenhum dos limites pré-estabelecidos foi ultrapassado, conseqüentemente, a manobra é aconselhável e a enumeração é encerrada com sucesso, isto é, o RD conclui como verdadeira a premissa da REGRA 001, apresentando o Plano de Manobra:

Plano de Manobra

Manobras para isolar o bloco com defeito

- Abrir chave 32
- Abrir chave 39
- Fechar chave 32
- Abrir chave 40

Manobra para restabelecer a área escura

- Fechar chave 24

A figura 2.16. ilustra a configuração da rede em operação após aplicação do plano de manobra apresentado acima.

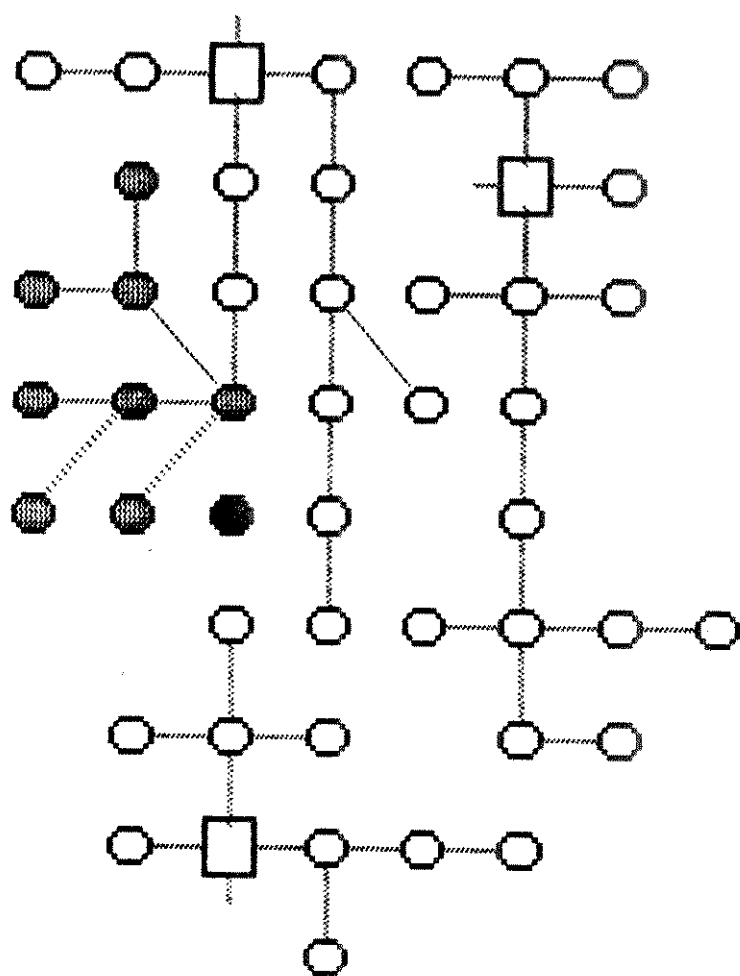


Figura 2.16. Rede em operação após aplicação do Plano de Manobra.

2.9. RESUMO E COMENTÁRIOS

Os pontos importantes discutidos neste capítulo foram:

- A representação do conhecimento como fatos na forma de triplas <Objeto,Atributo,Valor> e regras de produção na forma SE premissa ENTÃO ação.
- Distinção entre fato estático e sua instância dinâmica.
- A estrutura básica de consulta ao sistema RD conhecida como Árvore de Contexto.
- Distinção entre árvore de contexto estática e árvore de contexto dinâmica.
- Como é efetuado o raciocínio por contexto.
- A árvore de contexto específica do sistema RD e suas estruturas correlatas: tipo de contexto e atributos dos nós da árvore de contexto.
- As funções para avaliação das regras: função_p e função_a.
- Descrição detalhada do mecanismo de inferência do sistema RD.

Além dos tópicos acima, este capítulo contém a relação das regras de produção do RD e ilustra uma consulta ao sistema com um exemplo comentado.

No exemplo, note a necessidade de uma interface gráfica, que apresente a configuração da rede ao operador durante as várias etapas do raciocínio da máquina de inferência.

Um aspecto a destacar é o caso de uma manobra não ser realizada por limitação de carga (na subestação, no alimentador ou na chave de manobra). Neste caso, poderíamos subdividir a área escura em sub-árvores, o que poderia viabilizar o restabelecimento, parcial ou mesmo total, da área escura (agora como duas ou mais sub-árvores).

CAPÍTULO 3

SIMULADOR DE MANOBRA

Este capítulo apresenta um algoritmo baseado no Método dos Momentos para simular o comportamento de uma rede de distribuição de energia elétrica após o fechamento de uma chave - o Método dos Momentos (discutido no apêndice A) é conveniente pela sua simplicidade. A particularidade deste algoritmo é a utilização de um tratamento computacional baseado em conceitos de grafos, que leva à implementação de funções simples e de alta performance computacional.

Quando ocorre um defeito na rede de distribuição é necessário isolá-lo. Este procedimento separa a rede de distribuição em duas partes: rede energizada e área escura. A rede energizada compreende os elementos da rede que continuam em operação; a área escura é composta por elementos da rede que estão sem energia.

Durante o processo de decisão sobre qual chave deve ser fechada para o restabelecimento da energia à área escura, é importante prever os efeitos desta operação antes de efetivá-la. O algoritmo descrito neste capítulo verifica se a queda de tensão e a capacidade dos elementos de rede, que serão afetados pela manobra pretendida, estão dentro de limites previamente estabelecidos.

3.1. MODELO DA REDE DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Existem várias entidades de interesse na rede de distribuição. Define-se *bloco* (ou bloco de carga) como uma parcela da rede primária (13.8 kV) conexa sem possibilidades de seccionamento pela operação de chaves (dispositivos seccionadores e de proteção). O conjunto de blocos ligados por chaves fechadas e energizados a partir da mesma fonte (cubículo da subestação) denomina-se *alimentador*. Uma subestação possui um ou mais alimentadores e a rede de distribuição uma ou mais subestações.

O modelo de grafo é conveniente para representar as funções e relações entre as entidades da rede. No grafo da figura 3.1, os nós representam os blocos de carga, exceção feita aos nós R e S_n ($n = 1, 2, \dots$), que representam o nó de fechamento da rede e as subestações, respectivamente. Os arcos representam chaves fechadas ou abertas.

Neste modelo é possível distinguir a rede energizada, que compreende todos os elementos de rede (blocos e chaves fechadas) ligados entre si física e eletricamente. Na figura 3.2, os nós e arcos pertencem à rede energizada. As linhas tracejadas são chaves abertas, que definem as ligações físicas possíveis da rede de distribuição.

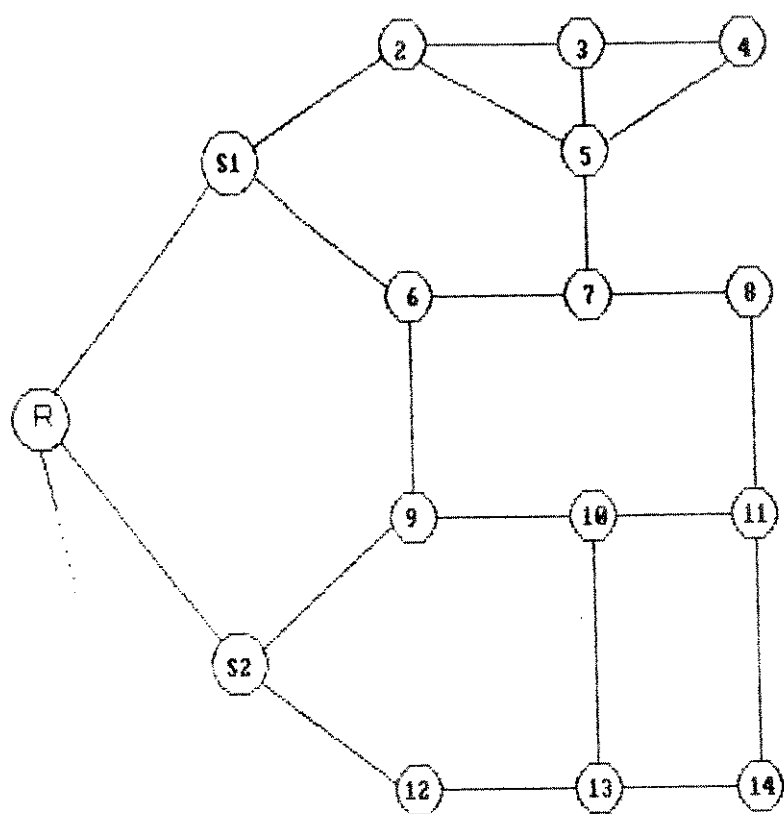


Figura 3.1. Modelo em grafo da rede de distribuição

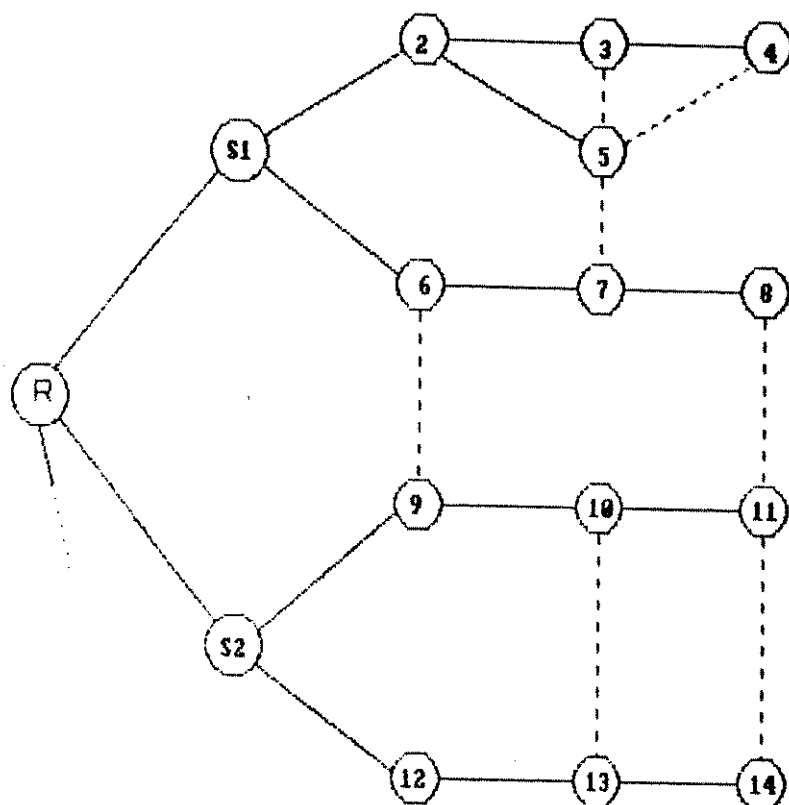


Figura 3.2. Representação da rede energizada no modelo de grafo

3.2. ALGUNS RESULTADOS MATEMÁTICOS

Seja o grafo $G(N,M)$, onde $N=\{1,2,\dots,n\}$ é o conjunto de nós (raiz, subestações e blocos) e o par ordenado $k = (i,j)$, para $i,j \in N$, o k -ésimo elemento do conjunto $M=\{1,2, \dots ,k, \dots ,m\}$ de arcos (chaves).

Define-se como rede energizada o grafo parcial $A(N, M')$, onde o conjunto $M' \subseteq M$ representa as chaves fechadas. A operação da rede de distribuição elétrica é radial, portanto A é uma árvore de G [KEN1].



Figura 3.3.

Sejam dois blocos $i, j \in N$ e uma chave $k \in M'$ (figura 3.3). O fluxo (potência ativa) na chave k é definido por:

$$f_k = p_j + \sum_{l \in \psi} p_l \quad (1)$$

onde

- p_j : carga própria do bloco j
- p_l : carga própria dos blocos descendentes de j
- ψ : conjunto dos blocos descendentes de j

A queda de tensão do bloco j , QT_j , pode ser obtida por (ver apêndice A):

$$QT_j = QT_i + K_k l_k f_k \quad (2)$$

Onde l_k é o "comprimento equivalente" (descrito no apêndice B) do bloco j atribuído à chave k , f_k é o fluxo (potência ativa) na chave k e a constante K_k é calculada a partir de parâmetros associados ao bloco j

$$K_k = \frac{r_k + x_k \operatorname{tg} \varphi}{\sqrt{3} |V_{op}|} \quad (3)$$

r_k : resistência por quilômetro de cabo (Ω/km)

x_k : reatância por quilômetro de cabo (Ω/km)

φ : ângulo de defasagem entre a tensão e corrente

V_{op} : tensão operativa (kV)

Fazendo $h_k = K_k l_k$ e substituindo em (2) temos:

$$QT_j = QT_i + h_k f_k \quad (4)$$

Observe que h_k é equivalente ao custo do arco em um modelo de fluxos em redes [KEN1].

Com o propósito de simplificar os cálculos, define-se potencial do bloco i (π_i) como

$$\pi_i = \sum_{k \in \Phi} K_k \cdot l_k = \sum_{k \in \Phi} h_k \quad (5)$$

onde Φ é o conjunto de chaves que pertencem ao caminho do bloco

l até raiz (bloco de fechamento R).

3.3. CÁLCULO DA QUEDA DE TENSÃO DEVIDA À VARIAÇÃO DE CARGA

Seja p'_i a carga própria de um bloco i , arbitrário, após uma variação de carga Δp , isto é

$$p'_i = p_i + \Delta p \quad (6)$$

Para toda chave $k \in \Phi$ (conjunto de chaves pertencentes ao caminho do bloco i até a raiz) o novo fluxo é obtido por:

$$f'_k = f_k + \Delta p \quad (7)$$

note que os fluxos nas demais chaves ficam inalterados.

Seja Ω o conjunto dos blocos pertencentes ao caminho do bloco i até a raiz. A nova queda de tensão, QT'_i , para todo bloco $i \in \Omega$ e a queda de tensão, QT'_j , para todo bloco j descendente (à esquerda e à direita) de $i \in \Omega$ pode ser calculada da seguinte forma:

$$QT'_i = QT_i + \Delta p \Pi_i \quad (8)$$

$$QT'_j = QT_j + \Delta p \Pi_i \quad (9)$$

Observe que a fórmula (9) somente é válida para o cálculo da queda de tensão em função de uma variação de carga, com os parâmetros físicos da rede inalterada.

3.4. ALGORITMO DE SIMULAÇÃO DE MANOBRA

A figura 3.4 ilustra o estado da rede de distribuição após uma contingência. Na figura, os blocos hachurados pertencem à rede energizada, o bloco com defeito (em preto) já foi isolado por manobras anteriores e a área escura está demarcada por um retângulo.

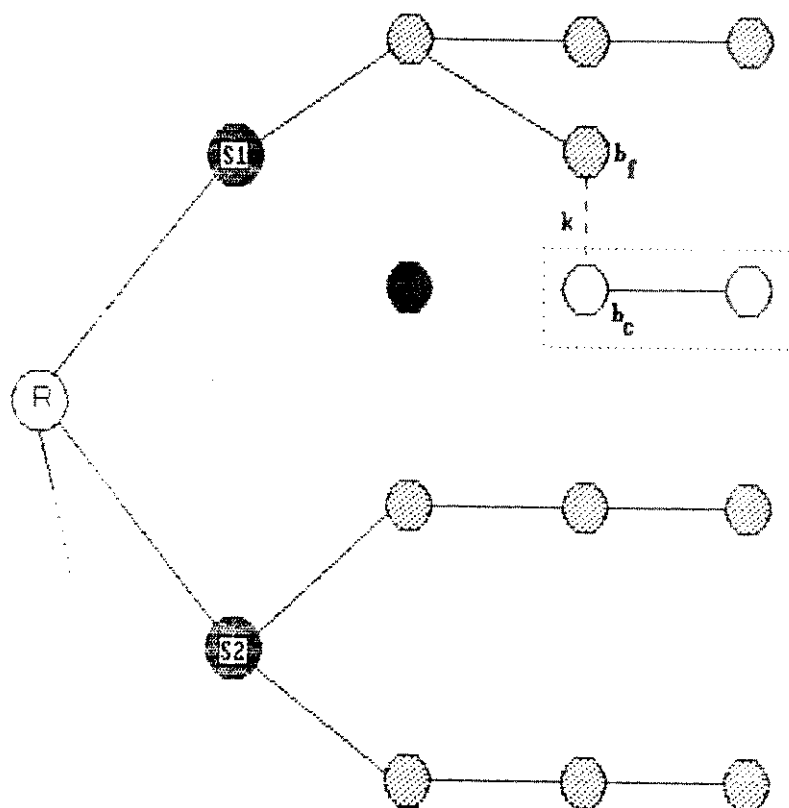


Figura 3.4.

Suponhamos que a chave $k \in (M - M')$ seja candidata a energizar a área escura e a carga a ser transferida seja Δp , que compreende o somatório das cargas próprias dos blocos pertencentes à área escura. O algoritmo descrito a seguir simula o efeito do fechamento da chave k na rede de distribuição. Como resultado, indica a adequação ou não da manobra pretendida. No algoritmo principal, SIMA, são feitas chamadas a algoritmos auxiliares, detalhados após a apresentação do algoritmo SIMA.

Algoritmo SIMA

Passo 1. Atribuições iniciais

- . Seja b_f o bloco fonte e b_c o bloco carga da chave k , para $f, c \in N$ e $k \in (M - M')$.
- . Faça capacidade = NO_LIMITE e
queda_tensão = NO_LIMITE

Passo 2. Avalie a capacidade da subestação suprir Δp

- . (Execute CAPSUB)

Passo 3. Avalie a capacidade física da chave k

- . (Execute CAPCHV)

Passo 4. Avalie a queda de tensão no bloco fonte (b_f).

- . (Execute QTBF)

Passo 5. Avalie a queda de tensão no bloco carga (b_c).

- . (Execute QTBC)

Passo 6. Avalie a queda de tensão nos blocos terminais da área escura.

- . (Execute QTAE)

Passo 7. Avalie a capacidade do alimentador do b_c suprir Δp de fluxo.

. (Execute CAPALIM)

Passo 8. Avalie a queda de tensão nos blocos terminais do alimentador do bloco b_c .

. (Execute QTALIM)

Se em qualquer uma das avaliações de queda de tensão ou capacidade os limites pré-estabelecidos forem ultrapassados o algoritmo retorna MANOBRA_NÃO_ACONSELHÁVEL. Caso contrário, o algoritmo retorna MANOBRA_ADEQUADA.

A seqüência de passos adotada no algoritmo SIMA é uma sugestão de encaminhamento. Critérios como esforço computacional, condições particulares da rede e outros podem ser adotados, obtendo-se ordenações diferentes para os passos do algoritmo.

Algoritmos Auxiliares

CAPSUB - Avalia a capacidade da subestação.

```

capsub ( )
. faça i = bc

/* busca a subestação */
[ enquanto tipo do bloco diferente de SE
  . faça i igual ao nó pai

/* verifica a capacidade da subestação */
. faça k arco pai do nó subestação

. faça fluxo =  $\frac{f_k + \Delta p}{1000 \cos \varphi}$ 

[ se fluxo > capk
  capacidade = FORA_DO_LIMITE
[ capacidade = NO_LIMITE

```

CAPCHV- Avalia a capacidade de chave de manobra

```

capchv ( )
. faça k igual a chave de manobra
/* converte fk em ampres */

fak =  $\frac{f_k}{\sqrt{3} |V_{op}| \cos \varphi}$ 

[ se fak > capk
  capacidade = FORA_DO_LIMITE
[ capacidade = NO_LIMITE

```

QTBF - Verifica a queda de tensão no bloco fonte, b_f .

```
qtbf ()  
.  
  faça  $QT'_{bf} = QT_{bf} + \Delta p \Pi_{bf}$   
  se  $QT'_{bf} > LIMITE\_SUPERIOR$   
  queda_tensão = FORA_DO_LIMITE  
  queda_tensão = NO_LIMITE
```

QTBC - Verifica a queda de tensão no bloco carga, b_c .

```
qtbc ()  
.  
  faça  $QT'_{bc} = QT_{bc} + \Delta p h_k$   
  se  $QT'_{bc} > LIMITE\_SUPERIOR$   
  queda_tensão = FORA_DO_LIMITE  
  queda_tensão = NO_LIMITE
```

QTAE - Percorre os blocos da área escura e verifica a queda de tensão em seus blocos terminais.

```
qtae ()  
.  
  faça  $\Delta QT = QT_{bc} - QT'_{bc}$   
  faça  $S$  o conjunto de blocos na sub-árvore  
  enraizada em  $b_f$   
  para todo  $i \in S$  faça  
  se  $b_i$  é terminal  
  . faça  $QT'_i = QT_i + \Delta QT$   
  se  $QT'_i > LIMITE\_SUPERIOR$   
  queda_tensão = FORA_DO_LIMITE  
  pare
```

CAPALIM - Avalia a capacidade do alimentador suprir a carga da área escura.

```

capalim ( )
. faça C o conjunto de blocos de bf até a
  subestação
  para todo l ∈ C faça
    . faça k arco pai de bl

    . faça  $\text{fluxo} = \frac{f_k + \Delta p}{\sqrt{3} |V_{op}| \cos \varphi}$ 

    se  $\text{fluxo} > \text{cap}_k$ 
      capacidade = FORA_DO_LIMITE
    pare
  capacidade = NO_LIMITE

```

QTALIM - Verifica a queda de tensão nos blocos terminais do alimentador.

```

qtalim ( )
. faça maxQT = 0
   $\Delta QT = \Delta p \times \Pi_{b_i}$ 
. faça C o conjunto de blocos de  $b_j$  até a
  subestação
  para todo  $i \in C$  faça
    se existir sucessor à direita de  $b_i$ 
    . seja D o conjunto de sucessores à
      direita de  $b_i$ 
      para todo  $j \in D$  faça
         $QT = QT_j + \Delta QT$ 
        se  $QT > \text{maxQT}$ 
        . faça maxQT = QT
    se existir sucessor à esquerda de  $b_i$ 
    . seja E o conjunto de sucessores à
      esquerda de  $b_i$ 
      para todo  $j \in E$  faça
         $QT = QT_j + \Delta QT$ 
        se  $QT > \text{maxQT}$ 
        . faça maxQT = QT
    . faça i apontar para o nó pai
       $\Delta QT = \Delta p + \Pi_i$ 
       $QT = QT_i + \Delta QT$ 
      se  $QT > \text{maxQT}$ 
      . faça maxQT = QT
  se maxQT > LIMITE_QT
  queda de tensão = FORA_DO_LIMITE
  queda de tensão = NO_LIMITE

```

3.6. ESTRUTURA DE DADOS

A escolha da estrutura de dados é uma das fases mais importantes no projeto de algoritmos, determinando sua eficiência e simplicidade de implementação. No SIMA a rede de distribuição é armazenada como uma estrutura de dados de grafo comumente denominada *Orientada por Arco* [JEN1].

Nesta estrutura de dados os elementos básicos de armazenamento são listas indexadas pelos arcos. Sejam $\Phi = \{o_k\}$ e $\Upsilon = \{t_k\}$ listas dos nós origem e término, respectivamente, de um grafo G . O k -ésimo elemento das listas Φ e Υ são os nós origem e término do arco k . A figura 3.5 ilustra o armazenamento nesta estrutura de dados.

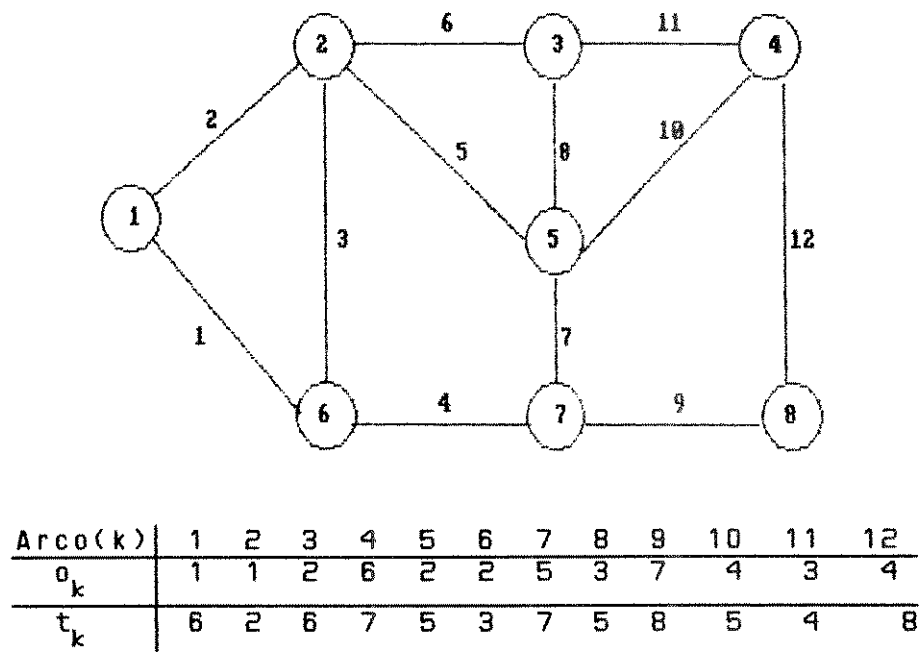


FIGURA3.5. Armazenamento do grafo na estrutura Orientada por Arco.

Os parâmetros pertinentes aos arcos são armazenados em listas do mesmo tipo. O espaço requerido por esta estrutura é $(\sum_{l=1}^p t_l) \cdot m$, onde p é o número de parâmetros (incluindo o nó início e fim), t_l é o tamanho em "bytes" do parâmetro e m é o número de arcos do grafo.

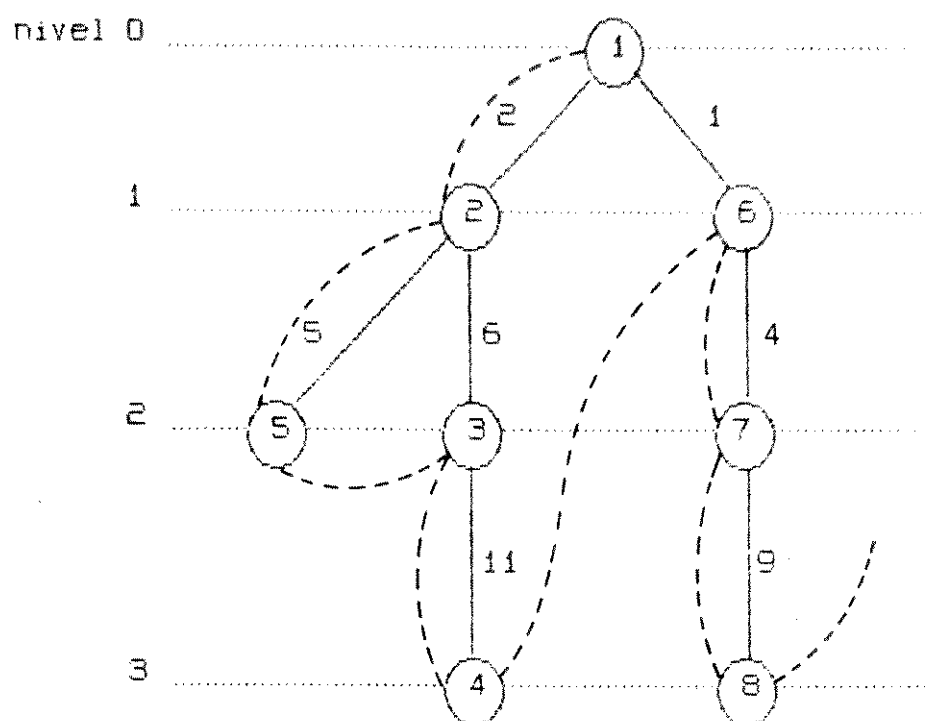
A árvore que representa a rede energizada pode ser armazenada adequadamente pela estrutura em preordem (filio) [JEN1, KEN1]. A estrutura em preordem usa três apontadores: P_B , P_F e P_P , para um nó i , arbitrário, da árvore temos:

$P_B(i)$: "back pointer" - apontador para o único arco que termina no nó i (arco pai).

$P_F(i)$: filio - apontador para o próximo nó quando a árvore é percorrida em preordem.

$P_P(i)$: pai - apontador para o nó pai.

A figura 3.6. exemplifica a estrutura em preordem.



Nó	1	2	3	4	5	6	7	8
P_B	0	2	6	11	5	1	4	9
P_F	2	5	4	6	3	7	8	-
P_P	0	1	2	3	2	1	6	7

FIGURA 3.6. Armazenamento da árvore na estrutura em preordem. As linhas tracejadas representam graficamente o fio, P_F , e as linhas pontilhadas o nível d_i .

A linha tracejada ligando os nós da árvore representa graficamente o apontador P_F .

Além desses apontadores, podem ser acrescentados à estrutura de dados rótulos, como, por exemplo, nível e

número de sucessores, objetivando aumentar a eficiência dos algoritmos. O uso ou não destes rótulos é um compromisso entre eficiência do algoritmo e espaço gasto na armazenagem. Em geral, a inclusão, na estrutura de dados, de uma ou mais listas para armazenar estes rótulos resultará uma redução do esforço computacional [KEN1].

A estrutura de dados implementada no SIMA é a preordem, acrescida do rótulo *nível* - número de nós da raiz a um nó *i* arbitrário. Esta estrutura possibilitou funções de atualização e busca eficientes e simples.

3.7. ASPECTOS DE IMPLEMENTAÇÃO DA ESTRUTURA DE DADOS

A implementação da estrutura de dados tratada no item 3.6 é descrita em detalhes a seguir. O arquivo de entrada contém os dados topológicos e elétricos da rede de distribuição. Os algoritmos de montagem da estrutura de grafo, da árvore, de tratamento e busca nas estruturas foram implementados como funções em linguagem C, objetivando os aspectos de eficiência, modularidade e portabilidade do código.

3.7.1. Arquivo de Entrada

O arquivo dos dados de entrada pode ser criado e/ou alterado com qualquer editor de texto que permita

gravação não formatada (arquivo ASCII). A figura 3.7. mostra o leiaute do arquivo.

1º Registro

número de nós
número de arcos

Um registro para cada nó contendo:

número do nó
tipo do bloco
carga própria do bloco
tipo de cabo no bloco
fator de potência no bloco
arco que liga ao pai

Um registro para cada arco contendo

número do arco
nó origem
nó término
capacidade do arco
tipo da chave
comprimento equivalente

Figura 3.7. Leiaute do arquivo de dados

Durante a leitura do arquivo de dados é feita a montagem da lista de arcos. Cada arco lido é incluído na lista em ordem crescente do nó origem. Para os nós que representam subestação (tipo do nó igual a 1) é incluído na lista um arco artificial do nó (Subestação) até a raiz (REDE), com capacidade do arco em MVA, representando a capacidade de suprimento de energia elétrica da subestação. Nos demais arcos (Chaves) a capacidade do arco é em ampere.

A estrutura de dados preordem da rede energizada é

montada a partir dos dados dos arcos que estão na base. A função `montre()`, apresentada a seguir, monta a árvore a partir do nó raiz, `nr`, da rede.

MONTRE - Monta Árvore

```
montre (nr)
┌
├   para todos os nós faça
├       fio do  $no_i$  =  $no_i$ 
├
├   para todos os nós faça
├       se existir arco pai para o  $no_i$ 
├           inclua o arco na árvore
├
└
```

3.7.2. Algoritmos Básicos

Os objetivos dos algoritmos básicos de tratamento e busca na estrutura de dados compreende inclusão e exclusão de um arco da árvore [JEN1], obtenção de uma subárvore enraizada no nó `i` [JEN1] e a recuperação do caminho do nó `i` à raiz [BRA1]. Quatro funções foram implementadas com estes objetivos.

INCTRE - inclui o arco k na árvore

```
inctre (k)
. Seja  $n_i$  e  $n_f$  os nós inicial e final,
  respectivamente, do arco k

. atualize o nível de  $n_f$ 

. faça aux =  $n_f$ 

  enquanto fio de aux  $\neq$   $n_f$ 
  . faça aux = fio de aux
  . atualize o nível de aux

. faça fio de aux      = fio de  $n_i$ 
  fio de  $n_i$           =  $n_f$ 
  arco pai de  $n_f$     = k
```

EXCTRE - Exclui o arco k da árvore

```
exctre (k)
. Seja  $n_i$  e  $n_f$  os nós inicial e final,
  respectivamente, do arco k

. faça aux =  $n_i$ 

  enquanto fio aux  $\neq$   $n_f$ 
  . faça aux = fio de aux

. faça aux1 = fio de  $n_f$ 

  enquanto o nível de aux > nível de  $n_f$ 
  . faça aux1 = fio de aux1

. faça fio de aux = fio de aux1
  fio de aux1 =  $n_f$ 

  /* enraiza a sub-árvore */
  arco pai de  $n_f$  = 0
```

RAIZ - obtem a sub-árvore enraizada no nó n_i .

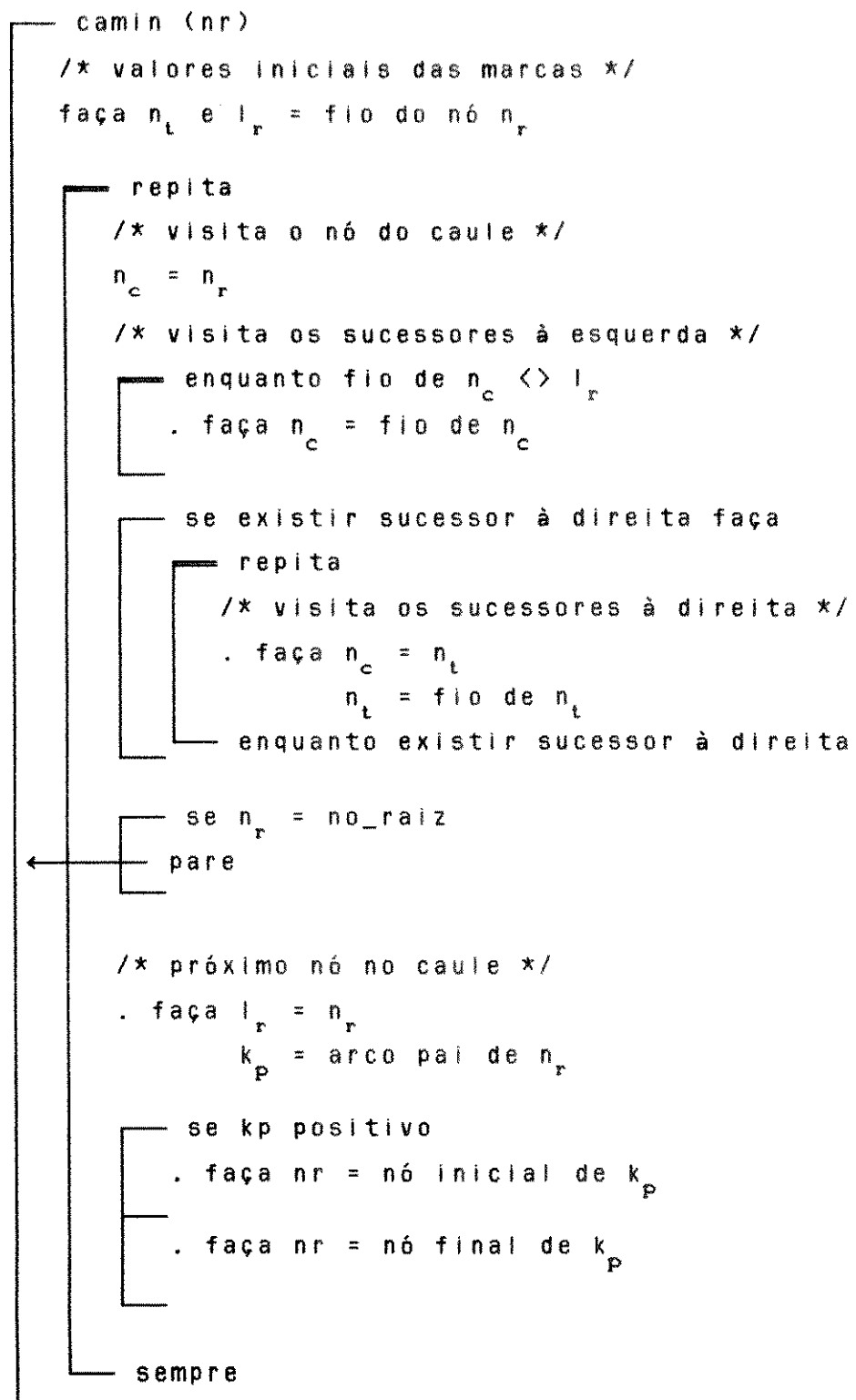
```
raiz (nó_raiz)
. Seja lista_nó lista de nós da subárvore

. Seja lista_arco lista de arcos da subárvore

. faça  $nó_i$  o fio do nó_raiz
  j = 1

  enquanto nível de  $nó_i$  < nível nó_raiz
  . faça lista_nó_j =  $nó_i$ 
    lista_arco_j = arco pai do  $nó_i$ 
     $nó_i$  = fio do  $nó_i$ 
    j = j + 1
```

CAMIN - percorre o caminho do nó n_l até a raiz



Note que o algoritmo CAMIN é colocado de forma genérica. Uma versão especializada é implementada a cada contexto que se aplica (ver algoritmo QTALIM no item 3.2.) Por exemplo, para se alterar o fluxo dos arcos do nó i , arbitrário, até a raiz, basta acrescentar o código adequado quando o próximo nó no caule é visitado.

CONCLUSÃO

O sistema especialista RD, proposto neste trabalho, foi desenvolvido para auxiliar o operador de rede na execução de planos de manobra emergenciais ou programadas em redes de distribuição de energia elétrica. Este sistema sugere planos de manobra baseados no conhecimento armazenado e em avaliações de capacidade de corrente e queda de tensão nos elementos que compõem a rede.

Um protótipo do sistema RD foi implementado em microcomputador compatível com IBM PC/XT. O protótipo integra módulos básicos de um sistema especialista (base de conhecimento, máquina de inferência e base de dados global) com módulos de computação numérica (gerenciador de rede e simulador de manobra). Esta integração apresenta uma arquitetura versátil para inclusões e/ou alterações de funções no sistema.

Durante o desenvolvimento do protótipo, várias máquinas de inferência foram experimentadas. Aquela que melhor se adaptou às necessidades foi baseada nas experiências do sistema especialista MYCIN e num desenvolvimento de idéias propostas por Sakaguchi e Matsumoto [SAK1] para restauração de sistemas de potência. A característica importante desta máquina de inferência é sua estratégia de controle, que além de ser "backward chaining" é direcionada por uma estrutura chamada árvore de contexto.

A árvore de contexto relaciona, hierarquicamente, as entidades (reais e conceituais) tratadas pelas regras de produção incluídas na base de conhecimento.

O algoritmo para simulação de manobras é baseado no Método dos Momentos (apêndice A), conveniente por sua simplicidade. Além disso, os resultados obtidos, no cálculo da queda de tensão ao longo do alimentador, são adequados para redes radiais de distribuição de energia elétrica. Este algoritmo tem a particularidade de utilizar conceitos de grafo no armazenamento e tratamento dos dados físicos e elétricos da rede. Conseqüentemente, o cálculo da queda de tensão nos blocos carga da rede, devido a uma alteração na carga, apresentou resultados computacionais extremamente eficientes.

A máquina de inferência, a base de conhecimento e a base de dados global estão implementadas em Turbo PROLOG 1.1. O gerenciador de rede, o simulador de manobra e a estrutura de dados foram implementados em Turbo C 1.5. Estes módulos interagem conforme o padrão de interfaceamento de linguagens fornecido pelo ambiente de programação.

O interfaceamento entre estas duas linguagens é relativamente fácil de ser implementada a nível de passagem de parâmetros atômicos (inteiro e ponto flutuante). No entanto, as informações e exemplos que constam dos manuais [BOR1, BOR2] não estão claros, dificultado muito a operacionalização deste interfaceamento.

A nosso ver, a utilização de duas linguagens tem vantagens e desvantagens. Como vantagens podemos listar:

- a) Os algoritmos de cálculo são programáveis eficientemente em C.
- b) A máquina de inferência, pelas suas características, tem sua programação facilitada no ambiente PROLOG.

Como desvantagens podemos citar:

- a) A necessidade de dominar duas linguagens de características totalmente diferentes.
- b) A memória compartilhada pelas duas linguagens deve ser alocada dinamicamente (esta técnica exige muitos cuidados e conhecimento minucioso dos compiladores C e PROLOG).
- c) A passagem de parâmetros não atômicos (listas e estruturas) não é trivial.

Os aspectos acima devem ser levados em consideração no momento da escolha do ambiente de programação.

Atualmente, a base de conhecimento do RD tem 13 (treze) regras básicas, o equivalente ao conhecimento de um operador com pouca experiência prática. A aquisição do conhecimento adicional de especialistas se dará, numa primeira fase, durante o desenvolvimento de um projeto piloto, a ser implantado em um dos 144 Centros de Operação

da Distribuição da Companhia Paulista de Força e Luz - CPFL (em discussão no momento da redação deste trabalho).

O sistema RD deverá funcionar como suporte de conhecimento (de um especialista) para operadores pouco experientes, diminuindo o fator falha humana. Para operadores experientes funcionará como "check list" do plano de manobras.

Como sugestão de trabalhos futuros e expansão do sistema RD fica:

- a) implementação da capacidade de explanação do raciocínio.
- b) desenvolvimento de uma máquina de inferência geral ("shell") com a característica de integrar módulos de computação numérica.
- c) exploração de outras formas de armazenamento e tratamento do conhecimento, aplicados nesta área do conhecimento humano.
- d) desenvolvimento de uma interface Homem/máquina que incorpore linguagem natural e recursos gráficos (p.ex. utilização de estação gráfica)¹.
- e) desenvolvimento de ferramentas para aquisição do conhecimento, o que facilitaria os trabalhos futuros.

¹ Já existe um protótipo desenvolvido separadamente para este fim.

- f) adaptação do sistema RD para ser utilizado como ferramenta de treinamento de operadores de rede (Sistema Tutorial Inteligente).
- g) Desenvolvimento de procedimentos por particionamento da área escura, caso não seja possível religá-la integralmente.

APÊNDICE A

A.1. MÉTODO DE CÁLCULO DE QUEDA DE TENSÃO

O método de cálculo da queda de tensão ao longo de um alimentador é função da precisão desejada nos resultados. No caso dos alimentadores primário de distribuição (13.8 kV) várias simplificações podem ser feitas, sem perdas significativas dos resultados obtidos. Para alimentadores de curta distância podemos desprezar a capacitância da linha, resultado apenas a impedância em série [MAR1].

A figura A.1. mostra o circuito e a figura A.2. o diagrama fasorial de uma linha de distribuição. A queda de tensão ΔV é dada por

$$\Delta V = V_F - V_C \quad (1)$$

na figura A.2. observa-se que ΔV é

$$\Delta V = \Delta V' + \Delta V'' \quad (2)$$

onde:

$$\Delta V' = I R_L \cos \varphi + I X_L \sin \varphi \quad (3)$$

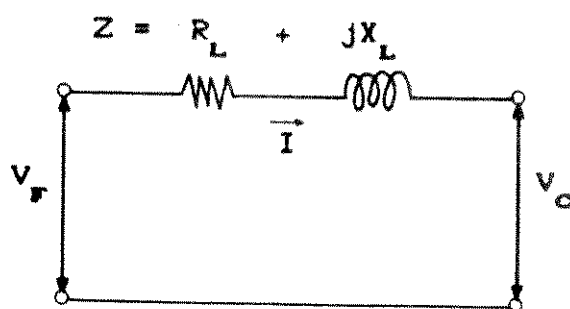


Figura A.1.

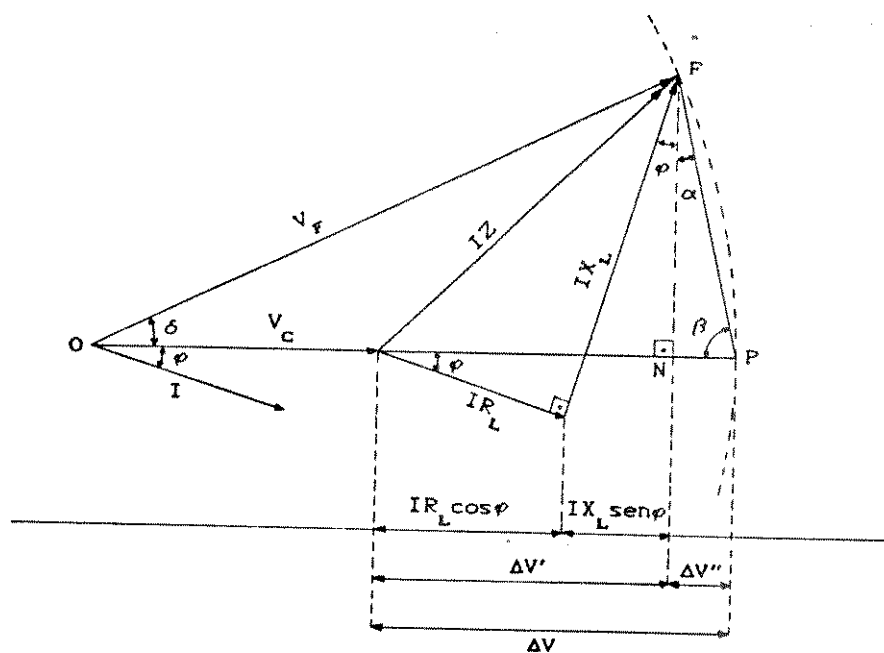


Figura A.2.

sendo $\overline{OFP}$ um triângulo isóscele temos

$$\beta = 90^\circ - \delta/2 \quad (4)$$

no triângulo $\overline{PNF}$

$$\alpha + \beta = 90^\circ \Rightarrow \beta = 90^\circ - \alpha \quad (5)$$

de (4) e (5) temos

$$\alpha = \delta/2 \quad (6)$$

então

$$\begin{aligned} \Delta V'' &= \overline{FP} \cos\beta = \\ &= \frac{\overline{FN}}{\cos\alpha} \cos\beta \\ &= \frac{V_F \sin\delta}{\cos\alpha} \cos\beta \\ &= \frac{2 V_F \sin\delta/2 \cos\delta/2}{\cos\delta/2} \cos(90^\circ - \delta/2) \\ &= 2 V_F \sin\delta/2 \sin\delta/2 = \\ &= 2 V_F \sin^2 \delta/2 \end{aligned} \quad (7)$$

retomando a equação (2) temos

$$\begin{aligned} \Delta V &= \Delta V' + \Delta V'' = \\ &= I R_L \cos\varphi + I X_L \sin\varphi + 2 V_F \sin^2 \delta/2 \end{aligned} \quad (8)$$

Analisemos agora a influência da parcela $2 V_F \sin^2 \delta/2$ na equação (8). Dividindo a equação (8) por V_F e expressando a queda de tensão em valores relativos

$$\frac{\Delta V}{V_F} = \frac{\Delta V'}{V_F} + 2 \sin^2 \delta/2 \quad (9)$$

A figura A.3. mostra a curva que representa o valor de $2 V_F \sin^2 \delta/2$ em função de δ .

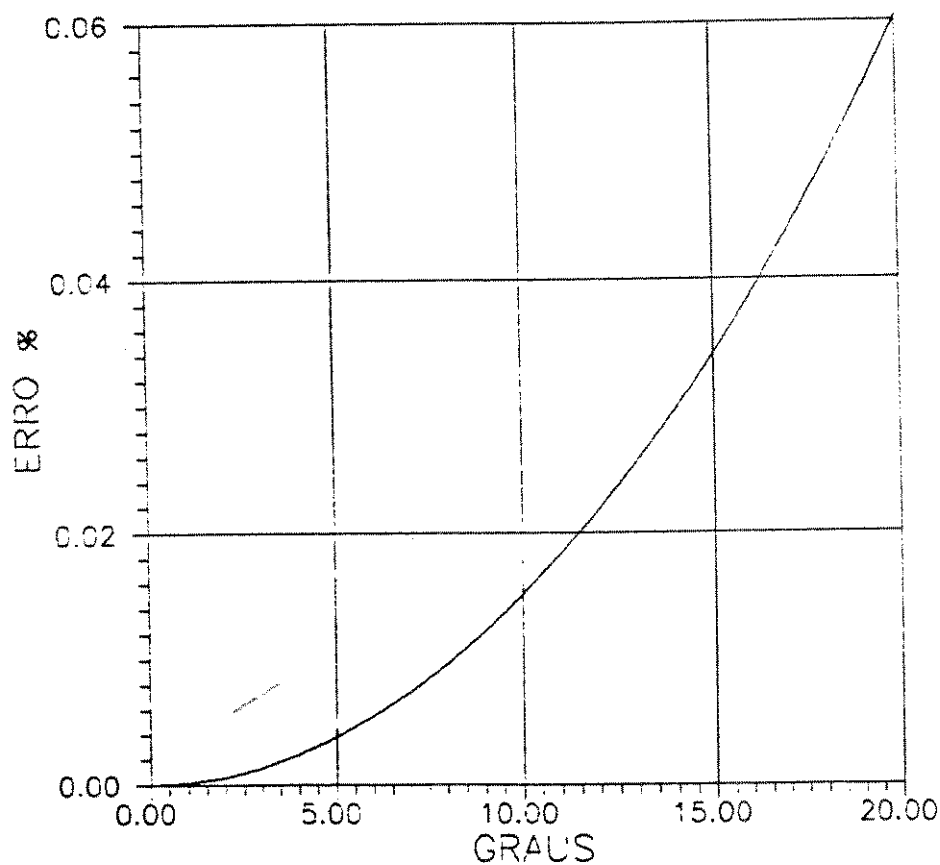


Figura A.3.

Note que para $\delta = 8^\circ$ o erro é da ordem de 1% e para δ da ordem de 3° o erro é 0,01%. Em alimentadores de distribuição as defasagens na prática são menores que 3° [SIGMAR], portanto, $2 V_F \sin^2 \delta / 2 \ll I R_L \cos \varphi + I X_L \sin \varphi$, o que justifica desprezar esta parcela. Assim, a queda de tensão é aproximadamente:

$$\Delta V \cong I R_L \cos \varphi + I X_L \sin \varphi \quad (10)$$

A.2. MÉTODO DOS MOMENTOS

Vários autores [MAR1, GON1, KAS1] descrevem o Método dos Momentos como um método para cálculo da queda de tensão em redes de distribuição de energia elétrica.

Como exemplo do método, suponhamos o circuito da figura A.1. seja um alimentador de comprimento l e a resistência e reatância da linha, R_L e X_L respectivamente, dada por

$$R_L = r \times l \quad (11)$$

$$X_L = x \times l \quad (12)$$

onde: r : resistência por quilômetro de cabo (Ω/km)
 x : reatância por quilômetro de cabo (Ω/km)

substituindo (11) e (12) em (10) temos

$$\begin{aligned}\Delta V &\cong I (r \times I) \cos\varphi + I (x \times I) \operatorname{sen}\varphi = \\ &= I I (r \cos\varphi + x \operatorname{sen}\varphi)\end{aligned}\quad (13)$$

multiplicando e dividindo a equação (13) por $\cos\varphi \frac{V_F}{\sqrt{3}}$ temos:

$$\Delta V \cong \frac{(r + x \operatorname{tg}\varphi)}{\sqrt{3} \frac{V_F}{V_S}} \sqrt{3} \frac{V_F}{V_S} I \cos\varphi I \quad (14)$$

$$\text{Seja } K = \frac{(r + x \operatorname{tg}\varphi)}{\sqrt{3} \frac{V_F}{V_S}} \text{ uma constante que depende}$$

do condutor usado, do fator de potência e da tensão de operação da rede. Substituindo na equação (14) temos:

$$\Delta V \cong K P I \quad (15)$$

O produto PI na equação (14) é conhecido como momento elétrico que a carga P exerce sobre a fonte. Considere um alimentador conforme ilustrado na figura A.4.

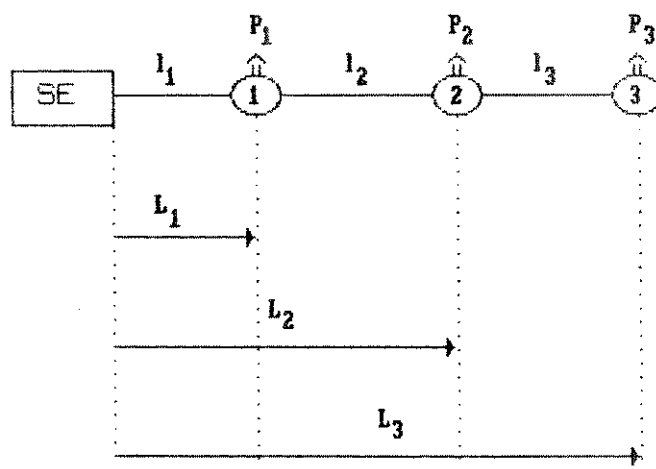


Figura A.4.

Suponhamos que o alimentador seja formado com condutores do mesmo tipo e as cargas tenham o mesmo fator de potência (essas condições não são restritivas do modelo):

$$\Delta V_1 \cong K (P_1 + P_2 + P_3) I_1 \quad (16)$$

$$\Delta V_2 \cong \Delta V_1 + K (P_2 + P_3) I_2 \quad (17)$$

$$\Delta V_3 \cong \Delta V_2 + K P_3 I_3 \quad (18)$$

Substituindo as equações (16) e (17) em (18) temos

$$\Delta V_3 \cong K \left[(P_1 + P_2 + P_3) I_1 + (P_2 + P_3) I_2 + P_3 I_3 \right] \quad (19)$$

Colocando a equação (19) em relação a L_n

$$\Delta V_3 \cong K \sum_{i=1}^3 P_i L_i \quad (20)$$

Observe que a queda de tensão , em (20), é o somatório dos momentos elétricos das cargas ao longo do alimentador.

APÊNDICE B

COMPRIMENTO EQUIVALENTE DE BLOCO DE CARGA

A aplicação do método dos momentos no cálculo da queda de tensão em um ponto arbitrário da rede requer a distância entre este ponto e o ponto anterior (ver apêndice A). No modelo da rede de distribuição adotado, os nós são blocos de carga e os arcos representam as chaves do sistema (Cap. 3.). A figura B.1. mostra a ampliação de um bloco de carga b_i .

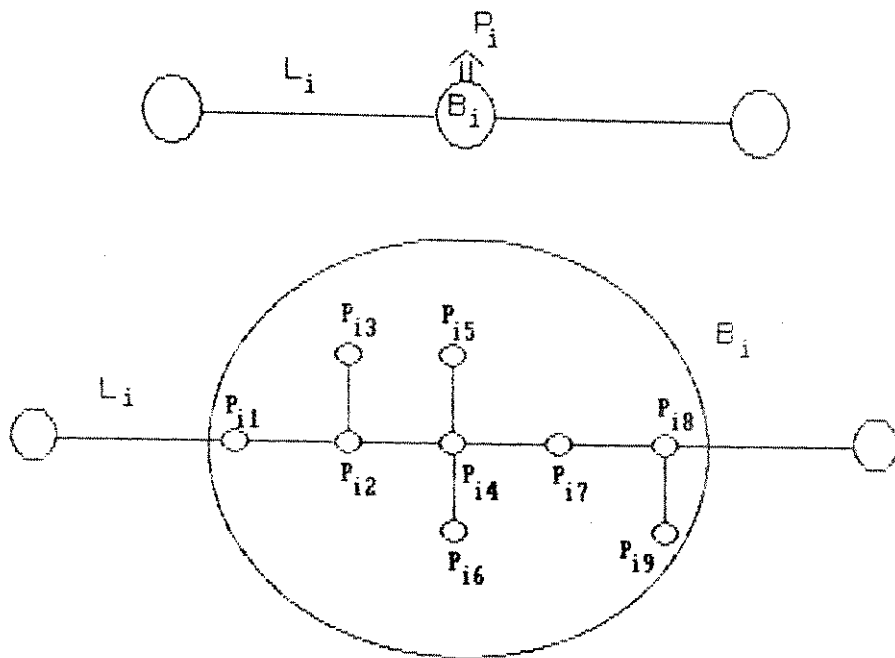


Figura B.1.

A demanda do bloco de carga é obtida pelo somatório das demandas dos transformadores. Denomina-se comprimento equivalente à distância calculada por um dos métodos:

- a) A maior distância compreendida entre a chave e um trecho terminal do bloco.
- b) O somatório dos comprimentos dos trechos ponderado pela carga própria do trecho.
- c) A distância compreendida entre a chave e o trecho terminal com maior carga própria.

O simulador de manobra, SIMA, trata o comprimento equivalente como um parâmetro constante definido para o bloco. Portanto, para o RD é indiferente o método de cálculo do comprimento equivalente adotado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BOR1] BORLAND. *TURBO C USER'S GUIDE*. 1987.
- [BOR2] BORLAND. *TURBO PROLOG OWNER'S HANDBOOK*. 1986.
- [BRA1] BRADLEY, G. W. & BROWN, G. G. & GRAVES, G. W. "DESIGN AND IMPLEMENTATION OF LARGE SCALE PRIMAL TRANSHIPMENT ALGORITHMS" *Management Science*, Vol. 24, #1, 1977. pp.1-34.
- [BUG1] BUCHANAN, B. G. & SHORTLIFE, E. H. *RULE-BASED EXPERT SYSTEM. The MYCIN experiments of the Stanford heuristic programming project*. Addison-Wesley, USA, 1985.
- [CAS1] CASTRO, C. H. & BUNCH, J. B. & TOPKA, T. M. "GENERALIZED ALGORITHMS FOR DISTRIBUTION FEEDER DEPLOYMENT AND SECTIONALIZING". *IEEE Transaction on Power Apparatus and System*, Vol. PAS-99, N. 2, March/April 1980. pp. 549-557.
- [CAST1] CASTRO Jr., G. A. *RECONFIGURAÇÃO AUTOMÁTICA DE REDES DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA*. Tese de mestrado apresentada à FEE/UNICAMP, Setembro 1985.
- [CPFL1] CPFL. "DESLIGAMENTO DE REDES E LINHAS DE DISTRIBUIÇÃO". *Orientação Técnica da Distribuição - OTD*. Código 501, Julho 1981.

- [CPFL2] CPFL. "CRITÉRIOS PARA ELABORAÇÃO DE PLANO DE MANOBRAS NA REDE DE DISTRIBUIÇÃO". Orientação Técnica Distribuição - OTD. Código 559.
- [CPFL3] CPFL. "IMPLANTAÇÃO DO CENTRO DE OPERAÇÃO DA DISTRIBUIÇÃO". Orientação Técnica da Distribuição - OTD. Código 709, Outubro 1982.
- [GON1] GONEN, T. *ELECTRIC POWER DISTRIBUTION SYSTEM ENGINEERING*. McGraw-Hill, USA, 1986.
- [HAR1] HARMON, P. & KING, D. *EXPERT SYSTEM. Artificial Intelligence in Business*. John Wiley & Sons, USA, 1985.
- [HARR1] HARRIS, M. E. R. *ALGORITMO PARA RECONFIGURAR REDES ELÉTRICAS DE DISTRIBUIÇÃO USANDO ABORDAGEM DE OTIMIZAÇÃO COMBINATÓRIA*. Tese de mestrado apresentada à FEE/UNICAMP, Dezembro 1987.
- [HAY1] HAYES-ROTH, F. & WATERMAN, D. & LENAT, D.B. *Building Expert System*. Addison-Wesley, USA, 1983.
- [JEN1] JENSEN, P. A. & BARNES, J. W. *NETWORK FLOW PROGRAMMING*. John Wiley & Sons, USA, 1980.
- [KAS1] KASATKIN, A. & PEREKALIN, N. *BASIC ELECTRICAL ENGINEERING*. MIR PUBLISHERS, MOSCOW, 1970.
- [KEN1] KENNINGTON, J. L. & HELGASON, R. V. *ALGORITHMS FOR NETWORK PROGRAMMING*. John Wiley & Sons, USA, 1980.

- [LIU1] LIU, C-C. & LEE, S. J. & VENKATA, S. S. "AN EXPERT SYSTEM OPERATIONAL AID FOR RESTAURATION AND LOSS REDUCTION OF DISTRIBUTION SYSTEM". IEEE Transaction on Power Systems, Vol. 3, N. 2, May 1988. pp. 619-625.
- [MAR1] MARCELIC, P. *LÍNEAS Y REDES ELÉCTRICAS*. EDIAR, ARGENTINA, 1957.
- [MAT1] MATTOS, M. G. *MODULO DE TREINAMENTO. Plano de Manobras*. CPFL.
- [MOR1] MORELATO, A. L. & MONTICELLI, A. "HEURISTIC SEARCH APPROACH TO DISTRIBUTION SYSTEM RESTORATION". 89 WM 111-6 PWRD.
- [RIC1] RICH, E. *ARTIFICIAL INTELLIGENCE*. McGraw-Hill, USA, 1983.
- [SAK1] SAKAGUCHI, T. & MATSUMOTO, K. "DEVELOPMENT OF KNOWLEDGE BASED SYSTEM FOR POWER SYSTEM RESTORATION". IEEE Transaction on Power Apparatus and System, Vol. PAS-103, # 2, February, 1983. pp 320-329.
- [SCH1] SCHNEIDER, H. M. & MARQUES, M. L. & COHN, P. G. "A INTELIGÊNCIA ARTIFICIAL E SUAS APLICAÇÕES NA AUTOMAÇÃO DA MANUFATURA". 2o. CONAI, 1985. pp. 500-509.
- [WIN1] WINSTON, P. H. *ARTIFICIAL INTELLIGENCE*, 2a edition. Addison-Wesley, USA, 1984.

[WOL1] WOLLENBERG, B. F. & SAKAGUCHI, T. "Artificial
Intelligence in Power System Operations".
Proceedings of the IEEE, Vol. 75, N. 12,
December 1987. pp. 1878-1885.