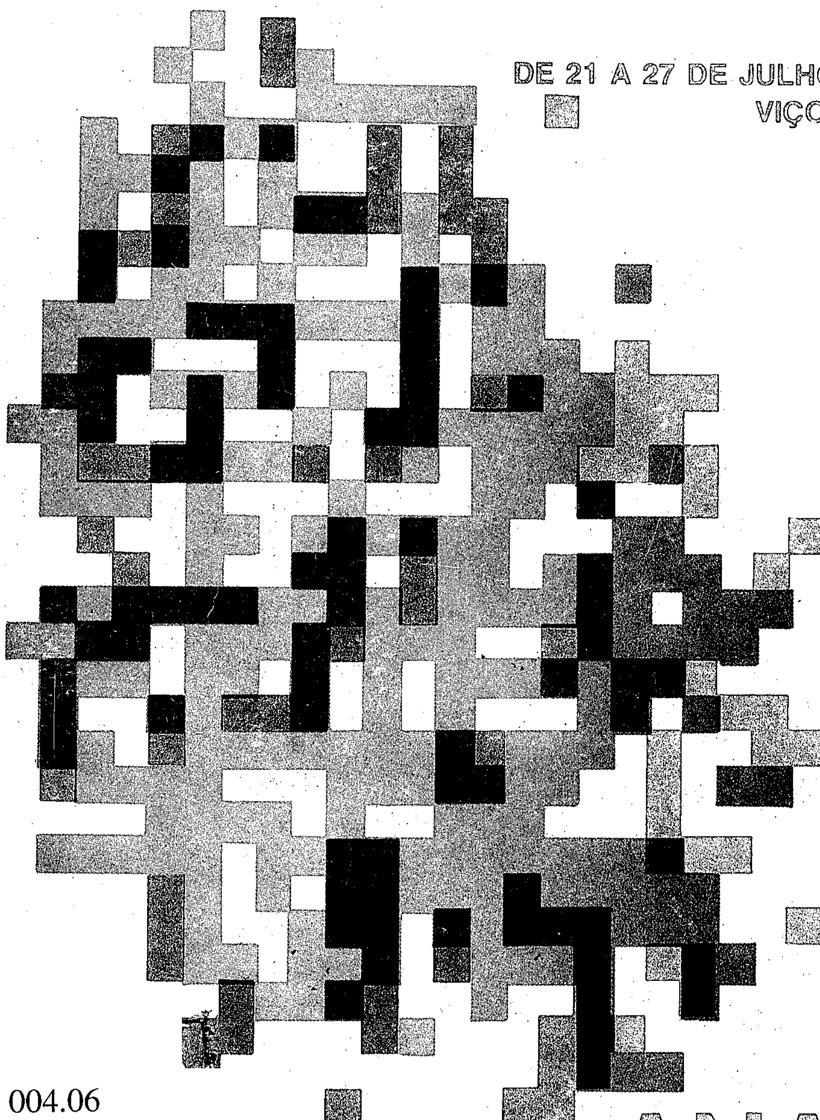
IV CONGRESSO

DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

DE 21 A 27 DE JULHO DE 1984

VIÇOSA — MG



ANAIS

VOL. I

IV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
VIÇOSA 21 a 27 de julho de 1984

ANAIS

VOLUME I

TRABALHOS APRESENTADOS

XI SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE

EDITORES:
R.S. BIGONHA, L.J. BRAGA-FILHO, A.M. OLIVEIRA & C.E. RECH

PROMOÇÃO: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV

PATROCÍNIO: CAPES CNPq CTI FINEP SEI

ESPECIFICAÇÃO DE PROTOCOLOS ATRAVÉS DE LÓGICA TEMPORAL - O PROTOCOLO DE ACESSO À BARRA DA REDE LOCAL - REDPUC.

L.F. GOMES SOARES * e D.A. MENASCÉ **

SUMÁRIO

Lógica temporal tem-se mostrado adequada para expressar uma série de propriedades da sequência de execução de eventos concorrentes, em particular protocolos. Este artigo trata justamente da especificação formal de protocolos usando este modelo matemático, apresentando como estudo de caso a especificação formal do protocolo de acesso à barra da rede local REDPUC.

ABSTRACT

Temporal logic has showed to be an useful tool to express a number of properties in the execution sequence of concurrent events, as is the case of protocols. This paper deals with the formal specification of protocols through the use of that mathematical model, giving the formal specification of the access protocol to the REDPUC local area network as a case study.

* Engenheiro Eletrônico (PUC/RJ, 1976), Mestre em Ciências em Engenharia Elétrica (PUC/RJ, 1979), Doutor em Informática (PUC/RJ, 1983); sistemas distribuídos, redes de computadores, automação de escritórios; professor do DEE - PUC/RJ, Rua Marquês de S. Vicente, 225 - Gávea - Rio de Janeiro, RJ - 22453.

** Engenheiro de Sistemas (PUC/RJ, 1974), Mestre em Ciências em Informática (PUC/RJ, 1975), Ph.D (UCLA , 1978); redes de computadores, banco de dados, automação de escritórios; professor do DINF - PUC/RJ, Rua Marquês de S. Vicente, 225 - Gávea - Rio de Janeiro, RJ - 22453.

1. INTRODUÇÃO

Lógica temporal é um ramo especial da lógica que trata do progresso de eventos no tempo, ao contrário da lógica clássica que só é adequada na descrição de situações estáticas [1 a 5].

A execução de um protocolo é precisamente uma cadeia de situações, chamadas estados de execução, que sofrem uma série de transformações determinadas pelas regras do protocolo. Lógica temporal mostra-se uma ferramenta adequada para modelagem de tal situação permitindo formalizar toda a execução do protocolo e não apenas as funções que ele controla. Este artigo trata justamente da especificação formal de protocolos usando este modelo matemático, apresentando como estudo de caso o protocólo de acesso à barra (PAB) da rede local REDPUC [5,6].

A segunda seção faz uma breve introdução informal à lógica temporal proposicional. A terceira seção consta de uma especificação, também informal e resumida, do PAB, salientando apenas os detalhes mais relevantes à compreensão da especificação formal a ser apresentada. Apenas algumas partes do PAB são especificadas com detalhes na quarta seção, visando não saturar o leitor. No apêndice, contudo, encontra-se a especificação formal de todo protocolo. A quinta seção é reservada às conclusões.

2. LÓGICA TEMPORAL PROPOSICIONAL - PRELIMINARES

Lógica temporal tem como domínio de interpretação um conjunto de estados e relações de acesso ligando estes estados. Restrições, no entanto, são impostas ao modelo de interpretação na estrutura da lógica temporal. Nesta, um estado s_2 é accessível a partir de um estado s_1 somente se, através do transcorrer do tempo, s_1 puder mudar para s_2. A história da progressão de estados é linear e discreta. Desta forma os modelos de lógica temporal consistem de sequências infinitas de estados $c=s_o,s_1,\ldots$ nas quais s_i é accessível a partir de s_j se e somente se $i \geq j$.

Dois operadores temporais são introduzidos na lógica clássica pela lógica temporal: são eles $\Box$ (chamado operador de necessidade) e $\Diamond$ (chamado de operador de possibilidade). Estes operadores são interpretados da seguinte forma:

$\Box$ w é verdade em um estado s_i se a fórmula w é verdade em todos os estados s_j, $j \geq i$

$\Diamond$ w é verdade em um estado s_i se a fórmula w é verdade em pelo menos um estado s_j, $j \geq i$.

Devido ao fato da sequência de estados ser discreta, pode-se neste caso falar não só de estados futuros, mas também de um único estado futuro imediato ao presente, ou próximo estado. Isto leva à introdução de um operador adicional, O, chamado próxima instância.

Devido ao fato do conjunto de estados formarem uma sequência linear, pode-se também aqui falar a respeito da progressão do valor verdade de uma fórmula até um certo estado. Isto leva à introdução de um outro operador, $\mathcal{U}$, chamado operador "until".

O significado destes novos operadores é dado pelas regras de interpretação que se seguem:

Ow é verdade em um estado s_i se a fórmula w é verdade no próximo estado s_{i+1}.

$\alpha \mathcal{U} \beta$ é verdade em um estado s_i se a fórmula β for verdade em algum estado s_j, $j \geq i$, e enquanto β não for verdade a fórmula α for verdade em todos os estados s_k, $i \leq k < j$.

Uma definição mais formal de lógica temporal proposicional pode ser encontrada em [5], onde o sistema axiomático para linguagens temporais proposicionais é apresentado como a seguir:

Axiomas Lógicos:

P1 : $\Box\alpha \supset \alpha$

P2 : $\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$

P3 : $O \sim \alpha \equiv \sim O\alpha$

P4 : $\Box(\alpha \supset O\alpha) \supset (\alpha \supset \Box\alpha)$

P5 : $\Box\alpha \supset O\,\Box\alpha$

P6 : $O(\alpha \supset \beta) \supset (O\alpha \supset O\beta)$

P7 : $\alpha \mathcal{U} \beta \supset \Diamond\beta$

P8 : $\Box\alpha \supset \alpha \mathcal{U} \beta$

P9 : $\alpha \mathcal{U} \beta \equiv \beta \vee (\alpha \wedge O(\alpha \mathcal{U} \beta))$

Regras de Inferência

RI1 : (Substituição Uniforme). O resultado da substituição de qualquer símbolo proposicional uniformemente em uma tese por qualquer fórmula bem formada da linguagem temporal proposicional é também uma tese.

RI2 : (Modus Ponens). Se α e $\alpha \supset \beta$ são teses, então β também o é.

RI3 : (Necessidade). Se α é uma tese, então $\Box\alpha$ é uma tese.

3. O PROTOCOLO DE ACESSO À BARRA (PAB) DA REDE LOCAL-REDPUC

Desenvolvida no laboratório de redes de computadores da PUC/RJ, a REDPUC é uma rede local em barra comum utilizando como meio de transmissão um cabo coaxial, e como procedimento de acesso a passagem de permissão ("token passing"). A especificação detalhada da rede local, bem como do PAB podem ser encontradas nas referências [5,6 e 7].

No PAB o acesso à barra pelos módulos é cíclico. Cada módulo ao transmitir passa a permissão para o próximo módulo que deve ganhar acesso à barra, e assim sucessivamente até o último módulo do ciclo. O último módulo define então o próximo intervalo de tempo como intervalo de contenção, e também o próximo módulo (o primeiro) a ganhar o acesso assim que o intervalo de contenção (IC) termine.

Cada módulo conectado à rede só pode transmitir uma vez em um ciclo e sempre deve fazê-lo mesmo não tendo dados a transmitir, pois tem de passar a permissão. A ordem de transmissão no ciclo não tem nenhuma relação com o endereço físico da estação.

O intervalo de contenção serve para que o módulo que quer se conectar à rede faça aí o seu pedido de inserção. É chamado de intervalo de contenção,pois é o único intervalo no qual pode haver contenção no acesso à barra na ausência de falhas.

O acesso ao meio no PAB é baseado no conceito de lista de controle de ciclo (LCC), que é a visão local de um módulo da sequência lógica de transmissões. Cada estação da rede contém sua própria LCC que inclusive pode ser diferente da de outros módulos,devido a possíveis erros e/ou falhas de transmissão e/ou recepção.

Ao final de cada transmissão um intervalo de tempo é dado para que o próximo módulo (o que recebeu a permissão) comece a transmitir. Esgotado este tempo o PAB considera que houve uma omissão. Quando um módulo "i" sente que uma estação "j" se omitiu e/ou transmitiu com erro por um número "n" de vezes consecutivas, retira a mesma da LCC. Tanto em caso de omissão ou erro o próximo a ganhar acesso ao meio é calculado baseado na LCC.

Quando um módulo percebe "k" ciclos consecutivos sem receber a permissão ou então "m" recepções com erro e/ou omissões consecutivas, suspeita que está em falha e sai da rede, acionando mecanismos de recuperação.

Como já mencionado, o intervalo de contenção é o mecanismo através do qual se dá a inclusão de módulos em um ciclo já estabelecido. Sempre ativado pelo último módulo do ciclo, este intervalo pode terminar de três modos diferentes:

a) O tempo máximo de espera de transmissão se esgota sem que haja nenhum pedido de inserção. Neste caso o ciclo segue normalmente como o anterior.

b) Um único pedido de inserção ocorre. Neste caso o ciclo segue normalmente com a inserção do módulo no final das LCC's.

c) Dois ou mais módulos pedem inserção causando interferência. Neste caso, é executado um procedimento de reinicialização, como será visto, com todos os módulos ativos.

Qualquer módulo, ao querer se inserir na rede ou vindo de uma interferência no intervalo de contenção, espera por um intervalo de tempo igual ao número máximo dos intervalos de omissões que podem haver em um ciclo. Se durante este tempo não ocorrer nenhuma transmissão, o módulo cria sua LCC como se todos os módulos estivessem conectados à rede e, a partir daí, dá início ao ciclo como se fosse a vez do primeiro a transmitir. A cada omissão, e/ou erro, o módulo omisso, ou em erro, é retirado da LCC. A este procedimento dá-se o nome de inicialização/reinicialização, e ele termina imediatamente após a entrada no intervalo de contenção, passando-se então ao funcionamento normal do ciclo.

4. ESPECIFICAÇÃO FORMAL DO PAB POR LÓGICA TEMPORAL PROPOSICIONAL

A especificação do PAB por lógica temporal foi feita definindo-se uma teoria, chamada Teoria da Passagem de Permissão . No contexto deste artigo, uma teoria é um par ordenado $T = (L, A)$, onde L é uma linguagem temporal proposicional e A é um conjunto de fórmulas bem formadas de L , chamadas axiomas não lógicos de T .

4.1. A Linguagem

Seja o conjundo $E =_{Df} \{e^1, e^2, \ldots, e^N, IC\}$, onde a interpretação para cada elemento de E é uma estação da rede local ou o intervalo de tempo para pedidos de inserção na rede (IC). N é o número máximo de estações. No que se segue, usou-se a notação "$P(e_i)$, $e_i \in E$", para indicar um conjunto de N+1 símbolos proposicionais $P(e^1)$, $P(e^2), \ldots P(IC)$. Os símbolos proposicionais da linguagem, juntamente com a interpretação pretendida, são então assim definidos, para todo $e_i \in E$:

$TD(e_i) \rightarrow$ verdade SSS a estação e_i está transmitindo dados.

$TI(e_i) \rightarrow$ verdade SSS a estação e_i está transmitindo pedidos de inserção.

$ER(e_i) \rightarrow$ verdade SSS a estação e_i detectou um erro de recepção.

$TO(e_i) \rightarrow$ verdade SSS a estação e_i esgotou o tempo de espera de que alguma estação transmitisse.

$TM(e_i) \rightarrow$ verdade SSS a estação e_i esgotou o tempo de espera de que houvesse um ciclo (rede em operação).

$INTCONT(e_i) \rightarrow$ verdade SSS a estação e_i percebeu a entrada de um intervalo de contenção.

$TT(e_i) \rightarrow$ verdade SSS o tempo que a estação e_i esperava receber a permissão se esgotou.

$QE(e_i) \rightarrow$ verdade SSS a estação e_i quer entrar na rede.

$CL(e_i) \rightarrow$ verdade SSS a estação e_i já transmitiu o pedido inserção e está definindo a sua LCC.

$NC(e_i) \rightarrow$ verdade SSS a estação e_i pensa que está inserida no ciclo (na rede) e já tem sua LCC toda criada.

$EF(e_i) \rightarrow$ verdade SSS a estação e_i percebe que sofreu alguma avaria ou foi indevidamente retirada da rede.

$CC(e_i) \rightarrow$ verdade SSS a estação e_i acabou de executar a reinicialização e está em fase de definição de sua LCC.

$TR(e_i,e_j)$, $e_j \in E \rightarrow$ verdade SSS o tempo que a estação e_i deu para que a estação e_j transmitisse sem erros esgotou.

$PROX_{e_i}(e_j,e_k)$ $e_j,e_k \in E \rightarrow$ verdade SSS e_k é a próxima estação que a estação e_i acha que deveria transmitir depois da estação e_j.

$Perm_{e_i}(e_j)$ $e_j \in E \rightarrow$ verdade SSS a estação e_i acha que é a estação e_j quem tem a permissão para transmitir.

4.2. Axiomas não Lógicos

Definidos todos os símbolos proposicionais necessários à especificação, é formulado então um conjunto de axiomas que define as regras de execução do protocolo.

Com a finalidade de não saturar o leitor, apenas alguns axiomas não lógicos são apresentados nesta seção, de forma a exemplificar a especificação de um protocolo por lógica temporal. Ao formular cada axioma, foi inicialmente dada uma interpretação ao mesmo sob a luz da interpretação dos símbolos proposicionais definidos. No apêndice, ao final do artigo, é apresentada a formulação de todos os axiomas não lógicos da Teoria da Passagem de Permissão. Na referência [5] pode ser encontrada uma especificação completa do PAB com a interpretação pretendida para cada axioma.

A13: "Qualquer transmissão é feita em um intervalo de tempo finito"

Este axioma mostra bem a utilização do operador $\Diamond$. O que se tem de formular é que uma estação estando transmitindo, tem em algum estado futuro de parar de transmitir, o que é definido pelas fórmulas:

A13: a) $TD(e) \supset \Diamond \sim TD(e)$ para todo $e \in E$

b) $TI(e) \supset \Diamond \sim TI(e)$ para todo $e \in E$

Outro exemplo de aplicação do operador $\Diamond$ pode ser visto na formulação do seguinte axioma:

A15: "Não pode haver uma sequência de transmissões de tempo infinito sem que as estações, que não estão em falha, percebam um intervalo de contenção".

A15: $\sim EF(e) \supset \Diamond (INTCONT(e) \vee EF(e))$ para todo $e \in E$

Finalmente, o axioma a seguir estabelece que:

A19: "Uma estação ao querer entrar na rede deve transmitir um pedido de inserção ao perceber o intervalo de contenção".

Este axioma é um exemplo da aplicação do operador O. Em outras palavras, ele define que uma estação no estado QE e tendo INTCONT(e) como verdadeiro, ao acabar a transmissão em andamento $[(\sim(TD(e^1) \vee TI(e^1))) \wedge (\sim(TD(e^2) \vee TI(e^2))) \wedge \ldots \wedge (\sim(TD(e^N) \vee TI(e^N)))]$, deve em seguida (no próximo estado) enviar o pedido de inserção. Ou seja,

A19: $(QE(e) \wedge INTCONT(e) \wedge (\forall_{e_2 \in E} \sim (TD(e_2) \vee TI(e_2)))) \supset O(TI(e))$

5. CONCLUSÃO

Uma pergunta que surge naturalmente é porque um sistema deve ser modelado formalmente. O objetivo primeiro é meramente descritivo. Outro objetivo é o da análise do modelo para a detecção de propriedades desejáveis e indesejáveis do sistema modelado. Desta forma o sucesso de um modelo é devido tanto ao seu poder de modelagem quanto ao seu poder de decisão.

Poder de modelagem diz respeito à capacidade que um modelo tem de representar fielmente as características de um sistema modelado. Poder de decisão diz respeito à capacidade que um modelo tem de determinar propriedades de um sistema modelado. Infelizmente, estes fatores são em geral dependentes, de forma que o aumento do poder de modelagem, isto é, da quantidade de sistemas que podem ser modelados, geralmente reduz a capacidade de determinação algorítmica dos modelos, tornando várias questões indecidíveis.

Lógica Temporal é bem conveniente na análise de questões específicas que envolvem características de estados particulares do sistema e não do sistema como um todo, por exemplo, se um estado certamente ocorrerá no futuro dado um estado presente, etc.

Embora o artigo não tenha tratado, por falta de espaço, da análise de algumas questões, é fácil ver que estas muitas vezes podem ser colocadas como simples teoremas, cuja demonstração prova sua validade. Por exemplo, questões como: "Uma estação sem falha ao querer entrar na rede sempre terá oportunidade de fazê-lo ou através de um pedido de inserção ou da execução do procedimento de reinicialização", poderia abstratamente ser colocado como: $QE(e) \supset \Diamond(TI(e) \vee CC(e) \vee EF(e))$, que é facilmente deduzível a partir dos axiomas A15 e A19.

Resta mencionar que o simples exercício de especificação formal de um protocolo já é suficiente para que a grande maioria de seus erros sejam detectados. Este fato já mencionado em experiências de outros grupos, também foi constatado no desenvolvimento do PAB pelo grupo de redes da PUC/RJ.

O PAB foi especificado formalmente por quatro modelos: dois por lógica temporal, um por redes de Petri com temporização, e outro por redes de Petri com predicados. Análise do protocolo utilizando estes modelos está sendo atualmente realizada, e espera-se, para breve publicação, que conclusões bem posicionadas possam ser tiradas a respeito da aplicabilidade de cada modelo, tanto para certas decisões quanto no refinamento de especificações de partes do protocolo.

APÊNDICE: AXIOMAS NÃO LÓGICOS DA TEORIA DA PASSAGEM DE PERMISSÃO

Definição: Seja α uma wff de L e $\mathcal{D}=\{a^1, a^2, \ldots a^z\}$

$$(1) \quad \underset{e_i \in D}{\top} \alpha(e_i) =_{Df} \alpha(a^1) \wedge \alpha(a^2) \wedge \ldots \wedge \alpha(a^z)$$

$$(2) \quad \sum_{e_i \in \mathcal{D}} \alpha(e_i) =_{Df} \alpha(a^1) \vee \alpha(a^2) \vee \ldots \vee \alpha(a^z)$$

$$(3) \quad \bar{\sum_{e_i \in \mathcal{D}}} (e_i) =_{Df} \sum_{e_j \in \mathcal{D}} (\alpha(e_j) \wedge \sim \sum_{\substack{e_i \in \mathcal{D} \\ e_i \neq e_j}} \alpha(e_i))$$

Para cada índice $e \in E$, $e \neq IC$, (ou seja, para cada estação) tem-se os seguintes grupos de axiomas não lógicos da teoria:

I - Axioma da Exclusão Mútua de Estados

Este axioma permite a uma estação estar em um e apenas um dos estados: QE CL,CC,NC, ou EF.

A1: a) $QE(e) \vee CL(e) \vee CC(e) \vee NC(e) \vee EF(e)$

b) Seja $S = \{QE, CL, CC, NC, EF\}$ para todo p, q em S , $p \neq q$

$$\sim(p(e) \wedge q(e))$$

II - Axiomas da Função do Próximo

Estes axiomas dizem respeito à estruturação, criação e manutenção (retirada e inserção de módulos) da lista de controle de ciclo pelas estações.

A2: a) $(NC(e) \vee CC(e)) \supset \prod_{e_i \varepsilon E} ((\bar{\sum_{e_j \varepsilon E}} \; Prox_e(e_i, e_j)) \wedge (\bar{\sum_{e_j \varepsilon E}} \; Prox_e(e_j, e_i)))$

b) $(NC(e) \vee CC(e)) \supset \sum_{\substack{e_i \varepsilon E \\ e_i \neq IC}} Prox_e(e_i, IC)$

Definição: (1) $INVPROX_e(f) =_{Df} \prod_{g \varepsilon E} (Prox_e(f,g) \supset O(Prox_e(f,g)))$

(2) $ELIMINE_e(f) =_{Df} \prod_{g \varepsilon E} \prod_{h \varepsilon E} ((Prox_e(g,f) \;\; Prox(f,h)) \supset$
$\supset O \; Prox_e(g,h))$

(3) $Prox^*_e(f, IC)$ é verdade SSS $f \; \varepsilon \; \{e_1, \ldots, e_k\}$ onde
$Prox_e(e_1, e_2) \wedge Prox_e(e2, e_3) \wedge \ldots \wedge Prox_e(e_k, IC)$

A3: Seja $A(e_1) \equiv NC(e) \wedge INTCONT(e) \wedge TI(e_1) \wedge O(\sim TI(e_1)) \wedge \sim ER(e)$

Para cada índice $e_1 \; \varepsilon \; E$, as wff's são axiomas:

a) $A(e_1) \supset (\sim Prox_e(e_1, e_1) \supset ELIMINE_e(e_1))$

b) $A(e_1) \supset (\prod_{f \varepsilon E} Prox_e(f, IC) \supset O(Prox_e(f, e_1)))$

c) $A(e_1) \supset O(Prox_e(e_1, IC))$

d) $A(e_1) \supset \prod_{\substack{e_2 \varepsilon E \\ e_2 \neq e_1}} (\sim(Prox_e(e_2, IC) \vee Prox_e(e_2, e_1)) \supset INVPROX_e(e_2))$

A4: Seja $A(e_1) \equiv (CC(e) \vee NC(e)) \wedge Perm_e(e_1) \wedge (ER(e) \vee TO(e)) \wedge O(\prod_{e_3 \varepsilon E} \sim(TD(e_3) \wedge TI(e_3)))$

Para cada índice $e_1 \varepsilon E$, $e_1 \neq e$, a wff é um axioma:

$(A(e_1) \wedge TR_e(e_1)) \supset (ELIMINE_e(e_1) \wedge \prod_{e_2 \varepsilon E, e_2 \neq e_1} (\sim(Prox_e(e_2, e_1) \vee (\sim Prox^*_e(e_2, IC))) \supset INVPROX_e(e_2)))$

A5: Seja $A \equiv \sum_{e_1 \in E} INTCONT(e) \wedge TI(e_1) \wedge O(\sim TI(e_1)) \wedge \sim ER(e)$

$B \equiv \sum_{e_1 \in E} (CC(e) \vee NC(e)) \wedge Perm_e(e_1) \wedge (ER(e) \vee TO(e)) \wedge (\prod_{e_2 \in E} O \sim (TD(e_2) \vee TI(e_2))) \wedge TR_e(e_1)$

$\prod_{e_3 \in E} (Prox^*_e(e_3, IC) \Rightarrow ((CC(e) \vee (NC(e) \wedge \sim A)) \wedge ((NC(e) \vee CC(e)) \wedge \sim B)) \supset INVPROX_e(e_3)))$

A6: Seja $A(e_1) \equiv TD(e_1) \wedge O \sim TD(e_1) \wedge \sim ER(e) \wedge \sim Prox_{e_1}(e_1, IC)$

Para cada índice $e_1 \in E$, as wff's são axiomas:

a) $(A(e_1) \wedge CL(e)) \supset ((\prod_{e_2 \in E} Perm_e(e_2) \supset (\prod_{e_3 \in E} Prox_{e_1}(e_1, e_3) \supset O\ Prox_e(e_2, e_3))) \wedge$

$\wedge (\prod_{e_4 \in E} (\sim Perm_e(e_4) \supset INVPROX_e(e_4))))$

b) $(\sim A(e_1) \wedge CL(e)) \supset \prod_{e_5 \in E} INVPROX_e(e_5)$

A7: Para cada índice $e_1 \in E$, a wff é um axioma:

$(QE(e) \wedge O(CL(e))) \supset (Perm_e(e_1) \supset (O(Prox_e(IC, e_1)) \wedge O(Prox_e(e\ , IC)) \wedge \prod_{\substack{e_2 \in E \\ e_2 \neq IC, e_2 \neq e}} INVPROX_e(e_2)))$

A8: $((QE(e) \vee CL(e) \vee NC(e)) \wedge O(CC(e))) \supset (O(Prox_e(IC, e^1) \wedge Prox_e(e^N, IC) \wedge \prod_{\substack{e^i \in E \\ e^i \neq IC, e^i \neq e^N}} Prox_e(e^i, e^{i+1})))$

onde i é o número de ordem da estação e^i.

III - Axiomas de Transmissão

Estes axiomas definem quando uma transmissão (de dados ou pedidos de inserção) pode ou deve ser feita, sua duração e a possível ocorrência de erros junto com sua detecção.

A9: $\sim(TD(e) \wedge TI(e))$

A10: $\sim(TD(IC) \vee TI(IC))$

A11: $TD(e) \supset (NC(e) \vee EF(e) \vee CC(e))$

A12: $TI(e) \supset (QE(e) \vee EF(e))$

A13: a) $TD(e) \supset \Diamond \sim TD(e)$

$TI(e) \supset \Diamond \sim TI(e)$

A14: $(\sum_{e_1 \in E} ((TD(e_1) \vee TI(e_1)) \wedge \sum_{\substack{e_2 \in E \\ e_2 \neq e_1}} (TD(e_2) \vee TI(e_2))) \wedge (\sim(TD(e) \vee TI(e)))) \supset ER(e)$

A15: $(NC(e) \vee CC(e) \vee CL(e) \vee QE(e)) \supset \Diamond(INTCONT(e) \vee EF(e))$

A16: $((NC(e) \vee CC(e)) \wedge Perm_e(e) \wedge (\prod_{e_1 \in E} \sim(TD(e_1) \vee TI(e_1))) \wedge (\sim INTCONT(e))) \supset O(TD(e))$

A17: a) $ER(e) \supset \sum_{\substack{e_1 \varepsilon E \\ e_1 \neq e}} (TD(e_1) \vee TI(e_1))$

b) $ER(e) \supset ER(e) \, U \, (\pi_{e_1 \varepsilon E} \sim (TD(e_1) \vee TI(e_1)))$

A18: $(TD(e) \vee TI(e)) \supset \sim ER(e)$

A19: $(QE(e) \wedge INTCONT(e) \wedge (\prod_{e_2 \varepsilon E} \sim (TD(e_2) \vee TI(e_2)))) \supset O(TI(e))$

A20: $\sum_{e_2 \varepsilon E} (TD(e_2) \vee TI(e_2)) \supset (\sim TO(e)) \wedge (\sim TM(e)))$

IV - Axiomas da Permissão do Acesso.

Estes axiomas dizem respeito a como as estações recebem e passam a permissão.

Definição: (1) $INVPERM_e =_{Df} \pi_{f \varepsilon E} (Perm_e(f) \supset O(Perm_e(f)))$

(2) $PERMPROX_e(f, g) =_{Df} \pi_{h \varepsilon E} (Perm_e(h) \supset Prox_f(g,h))'$

(3) $PROXPERM_e(f) =_{Df} \pi_{h \varepsilon E} (Perm_e(h) \supset Prox_e(h,f))$

A21: Seja $A(e_1) \equiv (NC(e) \vee CC(e)) \wedge \sim ER(e) \wedge TD(e_1) \wedge O \sim TD(e_1)$

$B \equiv (NC(e) \vee CC(e)) \wedge (ER(e) \vee TO(e)) \wedge \pi_{e_2 \varepsilon E} \; O \sim (TD(e_2) \vee TI(e_2))$

$C \equiv (NC(e) \vee CC(e)) \wedge \sim ER(e) \wedge \sum_{e_2 \varepsilon E} (TI(e_2) \wedge O \sim TI(e_2))$

$D \equiv (NC(e) \vee CC(e)) \wedge TD(e) \wedge O \sim TD(e)$

a) Para cada índice $e_1 \varepsilon E \quad e_1 \neq e$ tem-se:

$(A(e_1) \wedge Prox_{e1}(e_1, IC)) \supset O \; (PERMPROX_e(e_1, IC))$

b) Para Cada índice $e_1 \; \varepsilon \; E \quad e_1 \neq e$ tem-se:

$(A(e_1) \wedge \sim Prox_{e_1}(e_1, IC)) \supset O \; (PERMPROX_e(e_1, e_1))$

c) $(((B \wedge \sim INTCONT(e)) \vee C \vee D) \wedge \sim PROXPERM_e(IC)) \supset O(\pi_{e_2 \varepsilon E} \; Perm_e(e_2) \supset PERMPROX_e(e, e_2))$

d) $(((B \wedge \sim INTCONT(e)) \vee C \vee D) \wedge PROXPERM_e(IC)) \supset O \; PERMPROX_e(e, IC)$

e) Para cada índice $e_1 \; \varepsilon \; E \quad e_1 \neq e$

$((NC(e) \vee CC(e)) \wedge \sim (A \vee B \vee C \vee D)) \supset INVPERM_e$

A22: Seja $A(e_1) \equiv TD(e_1) \wedge O \sim TD(e_1) \wedge \sim ER(e) \wedge (QE(e) \vee CL(e))$

Para todo índice $e_1 \; \varepsilon \; E$ as wff's são axiomas:

a) $(A(e_1) \wedge \sim Prox_{e_1}(e_1, IC)) \supset O\ PERMPROX_e(e_1, e_1)$

b) $(A(e_1) \wedge Prox_{e_1}(e_1, IC)) \supset O\ PERMPROX_e(e_1, IC)$

c) $((QE(e) \vee CL(e)) \wedge \sim A(e_1) \supset INVPERM_e$

A23: $((QE(e) \vee CL(e) \vee NC(e)) \wedge O(CC(e))) \supset O\ Perm_e(e^1))$

onde e^1 é a estação de ordem 1.

V - Axiomas do Intervalo de Contenção.

Estes axiomas definem como as estações percebem a entrada em um intervalo de contenção

A24: Seja $D \equiv (NC(e) \vee CC(e)) \wedge TD(e) \wedge O \sim TD(e)$

$A(e_1) \equiv (NC(e) \vee CC(e)) \wedge TD(e_1) \wedge O(\sim TD(e_1)) \wedge \sim ER(e)$

$B(e_1) \equiv (NC(e) \vee CC(e)) \wedge TI(e_1) \wedge O(\sim TI(e_1)) \wedge \sim ER(e)$

$C \equiv (NC(e) \vee CC(e)) \wedge (ER(e) \vee TO(e)) \wedge O(\prod_{e_3 \in E} \sim (TD(e_3) \vee TI(e_3)))$

Para cada $e_1 \in E$, $e_1 \neq e$ as wff's são axiomas:

a) $(A(e_1) \wedge Prox_{e_1}(e_1, IC)) \supset O(INTCONT(e))$

b) $(A(e_1) \wedge \sim Prox_{e_1}(e_1, IC)) \supset O(\sim INTCONT(e))$

c) $((B \vee C \vee D) \wedge (\sim INTCONT(e)) \wedge PROXPERM_e(IC)) \supset O(INTCONT(e))$

d) $((B \vee C \vee D) \wedge (\sim INTCONT(e)) \wedge (\sim PROXPERM_e(IC))) \supset O(\sim INTCONT(e))$

e) $((B \vee C) \wedge INTCONT(e)) \supset O(\sim INTCONT(e))$

f) $((NC(e) \vee CC(e)) \wedge \sim(A \vee B \vee C \vee D)) \supset (((INTCONT(e) \wedge O(INTCONT(e))) \vee$
$\vee (\sim INTCONT(e) \wedge O \sim INTCONT(e)))$

A25: Seja $A \equiv \sum_{e_1 \in E} ((CL(e) \vee QE(e)) \wedge TD(e_1) \wedge O(\sim TD(e_1)) \wedge \sim ER(e) \wedge Prox_{e_1}(e_1, IC))$

a) $A \supset O(INTCONT(e))$

b) $((CL(e) \vee QE(e)) \wedge \sim A) \supset O(\sim INTCONT(e))$

6. AXIOMAS DA MUDANÇA DE ESTADO

Estes axiomas definem como as estações passam de um estado (NC,CC,CL, QE ou EF) a outro.

A26: Seja $A \equiv QE(e) \wedge TI(e) \wedge O(\sim TI(e))$

$B \equiv QE(e) \wedge TM(e)$

a) $A \supset O(CL(e) \vee EF(e))$

b) $B \supset O(CC(e) \vee EF(e))$

c) $(QE(e) \wedge \sim(A \vee B)) \supset O(QE(e) \vee EF(e))$

A27: Seja $A \equiv CL(e) \wedge \sum_{e_2 \in E} (TD(e_2) \wedge O \sim TD(e_2) \wedge \sim ER(e) \wedge Prox_{e_2}(e_2, e_1))$

$B \equiv CL(e) \wedge TM(e)$

$C \equiv CL(e) \wedge INTCONT(e)$

a) $A \supset O(NC(e) \vee EF(e))$

b) $B \supset O(CC(e) \vee EF(e))$

c) $C \supset O(QE(e) \vee EF(e))$

d) $(CL(e) \wedge \sim(A \vee B \vee C)) \supset O(CL(e) \vee EF(e))$

A28: Seja $A \equiv NC(e) \wedge TT(e)$

$B \equiv NC(e) \wedge TM(e)$

a) $A \supset O(QE(e) \vee EF(e))$

b) $B \supset O(CC(e) \vee EF(e))$

C) $(NC(e) \wedge \sim(A \vee B)) \supset O(NC(e) \vee EF(e))$

A29: a) $(CC(e) \wedge INTCONT(e)) \supset O(NC(e) \vee EF(e))$

b) $(CC(e) \wedge \sim INTCONT(e)) \supset O(CC(e) \vee EF(e))$

BIBLIOGRAFIA

[1] LAMPORT, Leslie. "What Good is Temporal Logic?" Information Processing 83, IFIP, 1983.

[2] HUGHES,G.E.; CRESSWELL,M.J. "An Introduction to Modal Logic", Methuen and Co LTD, London, 1968.

[3] MANNA,Z. ; PNUELI,A. "Verification of Concurrent Programs, Part I: The Temporal Framework", Technical Report nº STAN-CS-81-836, Stanford University, June 1981.

[4] SCHWARTZ,R.; MELLIAR-SMITH,P. "Temporal Logic Specification of Distributed Systems", Proceedings of the Second International Conference on Distributed Systems, Paris, France, April 1981.

[5] GOMES SOARES,L.F. "Projeto e Desenvolvimento de Protocolos para Redes Locais de Computadores", Tese de Doutorado, Departamento de Informática PUC/RJ, dezembro de 1983.

[6] GOMES SOARES,L.F; MENASCÉ,D.A. "Um Protocolo para Redes Locais do Tipo Difusão, Anales de la IX Conferência Latinoamericana de Informática, vol. 2, Panel Info'82, Lima-Peru, agosto de 1982