UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
INDUSTRIAL

DISSERTAÇÃO DE MESTRADO

Lógica Modal Aplicada à Verificação de Sistemas a Eventos Discretos

Autor: **Christiano Pereira Pessanha**

Orientador: **Prof. Dr. Rafael Santos Mendes**

Dissertação de mestrado submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do Título de Mestre em Engenharia Elétrica (Computação)

Banca Examinadora:
Rafael Santos Mendes - Orientador (FEEC / UNICAMP)
Fernando Antônio Campos Gomide (FEEC / UNICAMP)
Márcio Luiz de Andrade Netto (FEEC / UNICAMP)
Elias Humberto Alves (IFCH / UNICAMP)

Campinas 2004

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

P432L
P438L

Pessanha, Christiano Pereira
Lógica modal aplicada à verificação de sistemas a eventos discretos / Christiano Pereira Pessanha. -- Campinas, SP: [s.n.], 2004.

Orientador: Rafael Santos Mendes.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Teoria dos sistemas dinâmicos. 2. Lógica matemática não-classica. 3. Demonstração automática de teoremas. I. Mendes, Rafael Santos. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

RESUMO

Este trabalho objetiva o estudo da verificação de especificações em sistemas dinâmicos a eventos discretos via lógica modal. Um grafo de eventos temporizado (ou GET) é utilizado para modelar o sistema dinâmico que se deseja analisar. As relações entre as transições do GET e a especificação que se deseja verificar são expressas através de fórmulas da lógica modal NK. Um tableau analítico é utilizado para verificar se a especificação é conseqüência lógica do conjunto de fórmulas que representa a rede. Os ramos abertos retornados pelo tableau estão associados a modelos que falseiam a especificação. Introduz-se o conceito de terminação de um ramo e prova-se que a solução retornada por um ramo terminado está associada às características de minimalidade, causalidade, unicidade e impulsividade. Com base nesses teoremas propõe-se um algoritmo para o tableau. Descreve-se a implementação computacional do algoritmo. Exemplos de sua utilização são apresentados.

Palavras-Chave: Sistemas Dinâmicos, Sistema a Eventos Discretos, Lógica Modal, Tableau Analítico, Álgebra max-plus, Dióides.

ABSTRACT

This works is a study on the verification of specification in discrete event dynamic systems through the use of modal logic. Timed Event Graphs (TEG) described by its equations in dioids are used to model the system whose properties should be verified. NK modal logic formulas express the dynamics of the system as well as the specification to be verified. An Analytic Tableau is used to verify if the formula corresponding to the specification is a logical consequence of the formulas describing the system. The tableau will return open branches associated to models that falsify the specification. The concept of branch termination is introduced and proved to be associated to features as minimality, causality, unicity and impulsivity. A new algorithm, based on these concepts is proposed and its computational implementation is described. Examples are presented.

Key Words: Dinamic Systems, Discrete Event Systems, Modal Logic, Analitic Tableau, Max Plus Algebra, Dióids

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\triangle$: Fim de Definição

$\blacktriangle$: Fim de Teorema

$\mathbb{N}$: Conjunto dos Naturais

$D[\![z_1,\cdots z_p]\!]$: Dióide em p variáveis $z_1,...z_p$

$\oplus$: Soma no Dióide

$\otimes$: Multiplicação no Dióide

γ : Operador Atraso em Contagem

δ : Operador Atraso em Datação

$[a]$: Classe de Equivalência que Contém a $\in$ D

$\equiv$: Relação de Equivalência

$\sigma_{(i,j)}$: Prefixo (i, j)

$\models$: Conseqüência Lógica

$\vee$: Disjunção

$\wedge$: Conjunção

$\neg$: Negação

$\rightarrow$: Implicação

$\leftrightarrow$: Bi-implicação

CAPÍTULO 1

Introdução

Os sistemas dinâmicos podem ser classificados como contínuos ou discretos, sendo os Sistemas Dinâmicos a Variável Contínua (SDVC) usualmente descritos por equações diferenciais, caracterizados pelo seu espaço de estados contínuo e pelo mecanismo de transição de estado dirigido pelo tempo. As ferramentas matemáticas usualmente aplicadas na modelagem e controle de sistemas deste tipo são as equações diferenciais (e a diferenças). Porém essas ferramentas não são aplicadas com o mesmo sucesso em uma classe de sistemas dinâmicos denominados Sistemas dinâmicos a Eventos Discretos (SED). Esses sistemas estão associados à tecnologia do mundo atual como arquivos percorrendo redes de computadores, redes de comunicação, tráfego aéreo, rodoviário e ferroviário, linhas de manufatura dos mais diversos produtos como automóveis, autopeças, circuitos eletrônicos, produtos químicos, embalagens, etc. Os sistemas discretos caracterizam-se pelo espaço de estados discreto e pelo mecanismo de transição de estado dirigido pela ocorrência de eventos (justificando a nomenclatura de "sistemas a eventos discretos").

Não há ainda um consenso sobre o paradigma a ser adotado na modelagem dos Sistemas a Eventos Discretos, tendo-se atualmente muitas linhas de pesquisa como o controle supervisório proposto por Ramadge e Wonham (1989), as redes de Petri (Peterson, 1981 e Murata, 1989), a lógica temporal (Ostroff, 1991) e a álgebra de dióides (Bacelli et al., 1992).

As redes de Petri são uma maneira formal de descrever os Sistemas a Eventos Discretos, sendo as redes de Petri temporizadas utilizadas quando há interesse em se avaliar o desempenho do sistema discreto. Uma subclasse das redes de Petri denominada Grafos de Eventos Temporizados (GET) possui a característica de ter uma única transição de entrada e uma única transição de saída para todos os seus lugares. Uma única transição de saída indica que quaisquer possíveis conflitos pelos recursos foram previamente resolvidos enquanto uma única transição de entrada significa ausência de conflito pelos recursos no GET. Portanto, GET's podem ser utilizados como uma ferramenta para modelar a sincronização de Sistemas a Eventos Discretos onde não há concorrência ou compartilhamento de recursos.

A dinâmica de um GET pode ser descrita por equações que utilizam os operadores "max" e "+", onde o operador "+" está associado ao tempo de processamento de um ítem em cada lugar do GET enquanto o operador "max" está associado à sincronização da utilização dos ítens que chegam a uma

dada transição. As equações que utilizam esses dois operadores são não lineares numa álgebra convencional. Mas, se as operações de maximização e soma usual forem definidas como as operações $\oplus$ e $\otimes$, respectivamente, as equações de estado obtidas para o GET que se está modelando tornam-se lineares. A dinâmica de um GET pode ser descrita por esta álgebra não usual denominada (Max, +), pertencente a uma estrutura algébrica mais geral denominada dióide que possui como propriedade a idempotência da operação de adição (que implica na inexistência da operação inversa da adição). A álgebra de dióides, portanto, mostra-se estreitamente relacionada às redes de Petri, particularmente aos GET's, sendo possível representar a dinâmica de um GET através de equações lineares na álgebra de dióides. A representação obtida para a dinâmica do GET possui uma forte analogia com as equações de estado contínuo da teoria convencional de sistemas, podendo-se abter a partir delas uma relação de entrada-saída (ou uma função de transferência). Das diversas soluções obtidas via dióide para a dinâmica de um dado GET interessa a menor solução, ou solução mínima, correspondendo esta solução ao menor tempo de disparo das transições do GET (seqüência mais rápida de disparos das transições do GET) (uma apresentação das redes de Petri e da álgebra de dióides será vista no capítulo 2). Interessa também que a solução seja impulsiva ou, que a(s) entrada(s) do GET possa ser vista como tendo disponível desde um tempo infinito um número ilimitado de recursos; interessando também que a solução seja causal (para uma definição precisa de causalidade veja capítulo 4).

Um dos objetos de estudo de maior interesse para os GET's é o da obtenção de controladores capazes de fazer com que a função de transferência de um GET controlado satisfaça uma determinada especificação. A teoria de residuação, que procura tratar do problema de se encontrar soluções extremas para inequações, mostra-se de grande importância no tratamento do problema de síntese de controladores, pois este problema pode ser visto como a busca da maior solução para um sistema de inequações num determinado dióide (Cottenceau, 1999). Uma estratégia para se obter o controle de um GET é a do modelo de referência, que consiste na utilização de um modelo cujo comportamento deverá ser aproximado pelo sistema que se deseja controlar, calculando-se a variável de controle de modo a que esta seja máxima, mas ao mesmo tempo respeitando-se as restrições do modelo de referência adotado. Conforme pode ser visto em Cottenceau, 1999 e Lüders, 2001 há restrições para o tratamento algébrico do problema da síntese de controladores. Torna-se, portanto, de grande interesse a busca por soluções alternativas ao tratamento algébrico que possam viabilizar a síntese de controladores para os casos em que o tratamento algébrico não se mostra aplicável.

Na busca por alternativas ao tratamento algébrico foi proposta por Magossi (1998) uma lógica

denominada lógica modal proposicional NK capaz de modelar sistemas passíveis de descrição via equações lineares na álgebra de dióides e, portanto, passíveis de descrição via GET's. Através da lógica NK é possível verificar especificações em grafos de eventos utilizando-se o mecanismo de prova lógica por refutação do tableau analítico. O tableau se mostra um método de prova com características desejáveis, pois não se utiliza do mecanismo de substituição de fórmulas mas apenas das subfórmulas extraídas via regras tableau das próprias fórmulas que se está utilizando na prova. O procedimento para a obtenção de modelos que falseiem uma especificação consiste em se expressar as relações entre as transições do grafo de eventos no qual se deseja verificar essa especificação através de fórmulas da lógica NK, fazendo-se o mesmo com a especificação. Em seguida verifica-se, via tableau, se a fórmula que expressa a especificação é conseqüência lógica do conjunto de fórmulas que representa o grafo de eventos. Caso a fórmula que expressa a especificação seja conseqüência lógica do conjunto de fórmulas associado ao grafo de eventos então a especificação é satisfeita, caso contrário é considerada insatisfeita. No caso de falseamento da especificação o tableau retorna um ramo aberto ou, que não possui contradições, correspondendo a um modelo para essa situação (capítulo 3). Na busca por estratégias alternativas à estratégia algébrica de controle, modelos obtidos via tableau para casos em que restrições algébricas impedem a obtenção de controladores via dióide poderiam ser utilizados para a obtenção de controladores para esses GET's e especificações.

O algoritmo originalmente proposto para o tableau analítico da lógica NK lhe permite retornar ramos abertos (ou modelos que falseiem a especificação) associados a soluções não-mínimas, assim como não-causais e não impulsivas. Soluções desse tipo mostram-se indesejadas pois a solução procurada deve ser mínima, causal e impulsiva. Os teoremas desenvolvidos no capítulo 4 tem a finalidade de proporcionar um método de decisão que permita ao tableau identificar os ramos associados a essas soluções indesejadas, descartando-as e, ao mesmo tempo, permitir ao tableau identificar os ramos associados a soluções que sejam mínimas, causais e impulsivas. Propõe-se com base nestes resultados um método de decisão, sistematizado num algoritmo proposto ao fim do capítulo capaz de evitar soluções indesejadas do tipo não-mínima, não-causal e não impulsiva. Na segunda parte desse trabalho descreve-se a implementação do algoritmo proposto numa linguagem computacional de alto nível, no caso a linguagem Java™, apresentando-se exemplos de verificações de especificações em grafos de eventos ilustrando-se o algoritmo proposto.

Este trabalho está organizado do seguinte modo. No capítulo 2 faz-se uma revisão dos principais conceitos relativos as redes de Petri, enfatizando-se os grafos de eventos temporizados (ou GET's),

seguindo-se uma apresentação da álgebra de dióides e exemplos de modelagem de GET's por essa ferramenta algébrica. No capítulo 3 faz-se uma revisão da lógica NK conforme apresentada em Magossi (1998) bem como do tableau semântico seguido de exemplos. No capítulo 4 apresentam-se as definições e os teoremas que permitirão o processo de decisão utilizado no algoritmo proposto ao fim do capítulo. O capítulo 5 apresenta os detalhes da implementação computacional do algoritmo proposto ao fim do capítulo 4, inicialmente uma descrição das classes definidas para a implementação do programa, em seguida o procedimento de instanciação dos objetos que estendem essas classes apresentado na forma de fluxogramas representando o funcionamento da implementação. O capítulo 6 apresenta exemplos de verificações de especificações em diferentes GET's apresentando as trajetórias de disparos das transições e os ramos retornados pelo tableau. O capítulo 7 apresenta a conclusão do trabalho, seguindo-se as referências bibliográficas e os anexos apresentados na seguinte seqüência: Anexo A com uma revisão a respeito do método de prova tableau e resultados referentes ao tableau proposto para a lógica NK demonstrados em Magossi (1998), anexo B apresentando os teoremas da lógica NK demonstrados por Magossi (1998) e anexo C com exemplos de verificação de especificações na implementação do tableau com arquivos de entrada e capturas de tela da saída da implementação.

CAPÍTULO 2

Redes de Petri e Dióides.

2.1 Introdução.

Neste capítulo será feito um breve estudo a respeito de redes de Petri e dióides. Uma apresentação geral de redes de Petri será feita inicialmente para que, a seguir, a definição de **grafo de eventos**, como caso particular desta, seja feita. Num segundo momento a álgebra de dióides e sua correlação com os grafos a eventos será apresentada.

2.2 Redes de Petri

Os modelos baseados nas redes de Petri podem ser encontrados na literatura nos mais diversos contextos graças à versatilidade que essa técnica de modelagem de sistemas proporciona, como a modelagem de sistema determinísticos, não determinísticos, síncronos, assíncronos, com ou sem concorrência. Diversas fontes bibliográficas podem ser utilizadas para um estudo aprofundado das redes de Petri. Foram tomadas como base para esta apresentação os textos de Cassandras (1999) e Peterson (1981) particularmente a notação e formalização se basearão em Cassandras.

Pode-se definir as redes de Petri em dois passos. Definindo-se o **grafo** (ou estrutura) associado à rede, composto de lugares, transições e arcos ligando lugares a transições ou transições a lugares, e acrescentando-se num segundo momento a esse grafo um **estado inicial**, o conceito de "disparo de transição" e a definição da maneira pela qual o estado se modifica na ocorrência do disparo de uma transição. Dessa forma, torna-se possível modelar a evolução dinâmica de um sistema.

Nas redes de Petri os eventos são associados a transições. Para que uma transição ocorra (ou "dispare") certas condições deverão ser satisfeitas, condições que podem ser verificadas ao se observar os lugares da rede que constituem a "entrada" (ou "input") da transição em questão. Após o disparo de uma transição um novo estado é alcançado, alterando as condições de disparo das transições da rede.

Definição 2.1
Redes de Petri

Uma rede de Petri é uma quádrupla (P, T, A, w) onde :

$P = \{p_1, p_2, \ldots p_n\}$ é um conjunto finito de lugares.

$T = \{ t_1, t_2, \ldots t_n \}$ é um conjunto finito de transições.

A é um conjunto de arcos, subconjunto do conjunto $(P \times T) \cup (T \times P)$.

w é uma função peso onde $w : A \rightarrow \mathbb{N}$. $\triangle$

De um modo geral pode-se relaxar a condição de conjuntos finitos para conjuntos contáveis.

Os conjuntos de lugares de entrada e de saída de uma transição t_j são dados por:

- conjunto dos lugares de entrada : $I(t_j) = \left(p_i \mid (p_i, t_j) \in A \right)$
- conjunto dos lugares de saída : $O(t_j) = \left(p_i \mid (t_j, p_i) \in A \right)$

Da mesma forma os conjuntos de arcos de entrada e de saída de um lugar p_i são dados por:

- conjunto dos arcos de entrada : $I(p_i) = \left(t_j \mid (t_j, p_i) \in A \right)$
- conjunto dos arcos de saída : $O(p_i) = \left(t_j \mid (p_i, t_j) \in A \right)$

A figura 2.1 ilustra a representação gráfica usual dos elementos de uma rede de Petri.

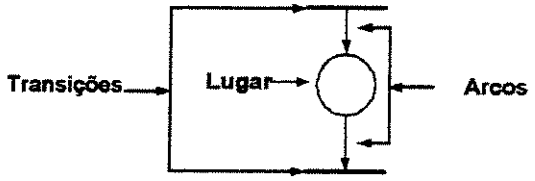


figura 2.1

Elementos de uma rede de Petri

Definição 2.2
Marcação.

Uma marcação x numa rede de Petri é uma função do tipo $x: P \rightarrow \mathbb{N}$, a qual associa a cada lugar p_i da rede de Petri um número inteiro não negativo $x(p_i)$ representando o número de "fichas" contidas em p_i (veja figura 2.2) . $\triangle$

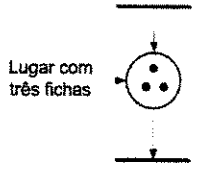


figura 2.2

Rede de Petri com três fichas

Logo, uma marcação determina um vetor $\mathbf{x} = [x(p_1), x(p_2), x(p_3), x(p_4), ..., x(p_n)]$ com dimensão igual ao número de lugares da rede. O "estado" de uma rede de Petri será dado pela sua marcação.

Definição 2.3
Rede de Petri marcada.

Uma rede de Petri marcada será notada por uma quíntupla $\langle P, T, A, w, x_0 \rangle$ onde $\langle P, T, A, w \rangle$ é uma rede de Petri, conforme definido anteriormente, e x_0 é uma marcação inicial. $\triangle$

Como foi dito anteriormente, as ocorrências dos eventos nas redes de Petri estão associadas a disparos nas transições sendo que uma idéia central em redes de Petri é utilizar a marcação da rede para determinar se uma transição pode ou não disparar. Para que um disparo ocorra numa transição é necessário que os lugares de entrada dessa transição contenham fichas em número suficiente (maior ou igual ao peso do arco que leva à transição).

Definição 2.4
Transição habilitada.

Uma transição $t_j \in T$, numa dada rede de Petri, está habilitada num estado $\mathbf{x} = [x(p_1), x(p_2), x(p_3), x(p_4) ..., x(p_n)]$ se e somente se $x(p_i) \geq w(p_i, t_j)$ para todo $p_i \in I(t_j)$. $\triangle$

Definição 2.5
Função de Transição de estado.

Numa rede de Petri marcada $\langle P, T, A, w, x_0 \rangle$ define-se a função de transição de estado $f: \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$ da seguinte maneira:

a) O domínio de f é restrito aos pontos $(\mathbf{x}, t_j)$ tais que a transição t_j esteja habilitada no estado $\mathbf{x}$.

b) $f(\mathbf{x}, t_j) = \mathbf{x}'$ onde $\forall i = i, ..., n / \ x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i)$. $\triangle$

Nota 1: Fixa-se assim o número de fichas dos lugares de entrada e saída em função da função peso do(s) arco(s) de entrada e saída da transição.

Nota 2: O número de fichas não se conserva necessariamente.

Um exemplo de transições de estado em uma rede de Petri é dado pela figura 2.3 abaixo:

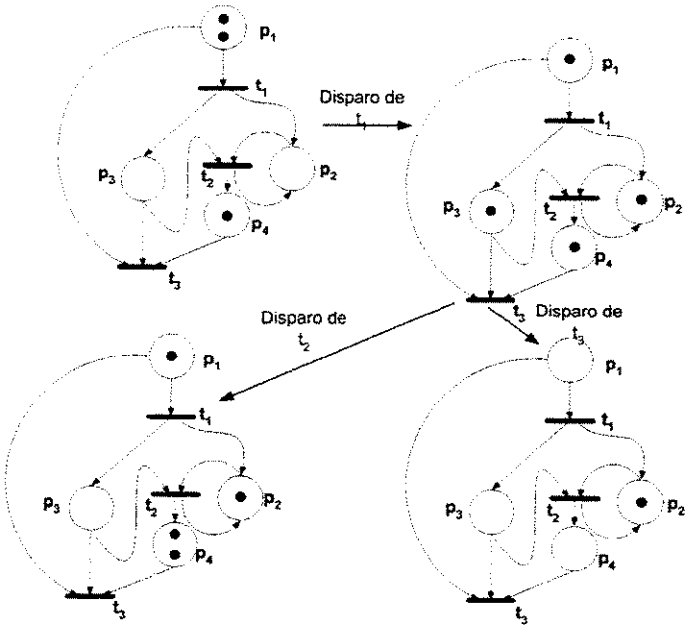


figura 2.3

Exemplos de disparos de transições em uma rede de Petri

Nota-se na figura 2.3 a dinâmica da rede de Petri através da movimentação de fichas ao longo dos lugares da rede gerando com isso uma modificação do conjunto de transições habilitadas que por sua vez disparam movimentando as fichas pelos lugares o que modifica novamente o conjunto de transições habilitadas e assim sucessivamente desde que a configuração da rede o permita, pois após um certo número de disparos de transições partindo do estado inicial a rede pode vir a ter apenas transições desabilitadas, no caso da rede da figura 2.3 a seqüência de disparos $t_1 \rightarrow t_3$ leva a rede a uma situação em que não há mais transições possíveis de serem disparadas (a rede estaria no estado terminal e nesse caso diz-se que a rede está "morta") enquanto que a seqüência $t_1 \rightarrow t_2$ não leva a esta situação pois a transição t_1 ainda pode disparar. Pode-se dizer que uma rede de Petri com dado estado inicial x_0 é **viva** se a partir de qualquer estado alcançado a partir de x_0 existir alguma trajetória na qual uma transição qualquer possa disparar (veja Cassandras, 1999 para mais detalhes sobre vivacidade). Uma rede de Petri com com dado estado inicial x_0 é dita conservativa em relação a um vetor $\gamma = \left[\gamma_1, \gamma_2, \gamma_3 \ldots . \gamma_n\right]$ onde $\gamma_i = \{0, 1\}$ se: $\sum_{i=1}^{n} \gamma_i x(p_i) =$ constante para qualquer trajetória possível. Uma rede de Petri conservativa pode representar um sistema onde os recursos não são criados nem destruídos.

2.3 Grafo de Eventos Temporizado

Um grafo de eventos é um caso especial das redes de Petri com as seguintes características:

I - Para cada lugar da rede, existe apenas uma transição de entrada como também uma transição de saída.

II – Todos os arcos da rede possuem peso 1 (arcos simples).

Definição 2.6
Grafo de Eventos

Um grafo de eventos é uma rede de Petri $\langle P, T, A, w \rangle$ onde :

$\forall(p_i \in P)$: card($I(p_i)$) = 1 e card($O(p_i)$) = 1

$\forall((t_i, p_j) \in A)$ então $w(t_i, p_j) = 1$ e $\forall((p_j, t_i) \in A)$ então $w(p_j, t_i) = 1$ Δ

Discutiu-se nos itens anteriores, os eventos e sua ordenação nas redes de Petri (dinâmica das redes), mas é de interesse o estudo dos grafos de eventos temporizados onde será associado um tempo de permanência nas marcações, concentrando-se a temporização nos lugares do grafo. O tempo será representado na forma de "barras" (ou de um número inteiro ao lado do lugar) inseridas junto às fichas nos lugares da rede caracterizando a **permanência mínima** das fichas nestes lugares. Assim fica caracterizado o tempo de espera, ou atraso para um dado lugar da rede, indicando o tempo mínimo que as fichas que chegam a este lugar devem "esperar" antes de contribuírem para a habilitação das transições de saída daquele mesmo lugar.Um grafo de eventos com p-temporização é chamado de grafo de eventos temporizado (GET). Na figura abaixo 2.4 tem-se um exemplo de um grafo de eventos temporizado.

A temporização do GET pode estar associada também às transições (conhecida como **t-temporização**) que após atingida a condição de habilitada deve "esperar" um dado número de unidades de tempo antes do disparo. A opção pela p-temporização não gera perda de generalidade nos GET's.

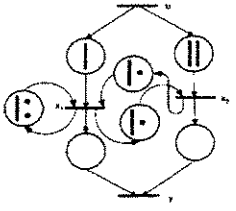


figura 2.4
Grafo de Eventos Temporizado

É importante destacar que, por definição, um grafo de eventos não permite situações de decisão, normalmente representadas em redes de Petri por múltiplos arcos entrando ou saindo dos lugares conforme representado na figura 2.5 abaixo. Esse fato permite concluir que os GET's constituem modelos para uma classe restrita de SED's.

figura 2.5
Rede de Petri com decisão

Além disso é importante destacar que os GET's considerados nesse trabalho possuem a característica de serem conectados, ou seja, a partir da transição de entrada pode-se chegar a qualquer outra transição do GET (existe sempre um "caminho" entre a entrada e qualquer outra transição do GET).

2.4 Trajetórias Associadas às Transições.

Consideremos um SED representável por um GET. Como foi dito anteriormente, os eventos de um SED (especificamente neste contexto, de um GET) estão associados aos disparos das transições do GET correspondente, com cada transição respectivamente associada a um dado evento. Pode-se representar a evolução de um dado evento no plano n×t (eventos × tempo). Cada ponto (i, j) pertencente a esse plano indicará que:

- No instante j ocorreram no máximo i disparos.

e

- O i-ésimo disparo ocorreu no mínimo no instante j.

Dessa forma todos os pontos à direita e abaixo do ponto (i, j) representarão situações impossíveis. Esta representação é escolhida porque deseja-se estabelecer uma álgebra que relacione as diversas informações disponíveis num GET e, dado que as conexões numa transição são inibitórias do disparo dessa transição, opta-se por representar os pontos associados às situações cuja ocorrência é inviável no GET. A concatenação de duas informações diferentes ocorrendo na mesma transição será portanto a união dos "cones a sudeste" (i.e. pontos à direita e abaixo) das duas informações dadas.

A trajetória de uma transição será representada por uma curva monotonamente crescente composta de segmentos horizontais e verticais (formando "degraus", ou cones com vértice nos pontos

de n×t, no plano) onde todos os pontos abaixo da curva representam situações impossíveis, e todos os pontos acima as situações possíveis. A curva propriamente representa a execução mais rápida possível para a trajetória da transição que está sendo representada no plano n×t. Neste contexto deve-se compreender um ponto no plano n×t como representante de todos os pontos abaixo e à direita dele (cone a sudeste do ponto) conforme mostra a figura 2.6 abaixo.

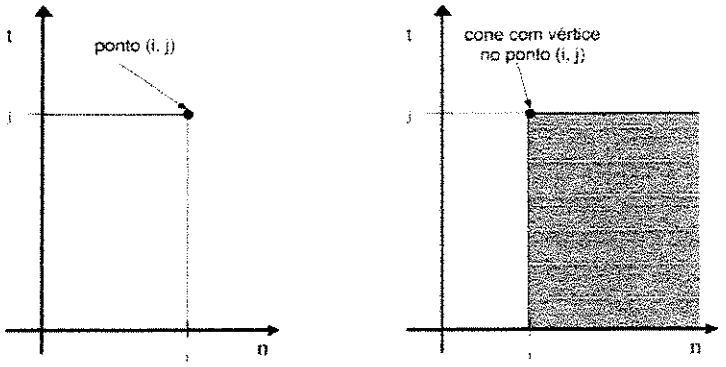


figura 2.6

Ponto (i, j) do plano n x t e cone com vértice no ponto (i, j)

Coloca-se portanto a seguinte questão: Supondo conhecida a trajetória de disparos numa certa transição de um GET (i. e. conhecidos alguns pontos do plano n×t associados a esta transição) como tal informação evolui ao longo da rede ?

A seguir serão descritas as duas formas de operação com os pontos no plano n×t de modo a representar as informações contidas nos lugares do grafo que chegam a uma de suas transições. Essa informação provém de arcos em situação de "entrada" na transição sendo que estes arcos estão em série ou em paralelo entre si, de modo que em cada uma das duas situações a informação que chega até a transição é tratada de maneira diferente.

Nas figuras abaixo tem-se um exemplo de como se compõe na transição de saída a informação contida nos lugares do grafo. Considere o grafo da figura 2.7 no qual assume-se que a transição u é disparada infinitas vezes em t = 0. Esta informação é representada pelo cone com vértice no ponto (0, 0) sendo esta situação análoga a uma entrada impulsiva em sistemas com dinâmica contínua. Neste grafo observa-se a situação em que as transições se encontram em "série", sendo a informação da transição de saída "y" dada pela soma vetorial dos pontos (2, 2) e (4, 3) no plano n×t resultando no ponto (2 + 4, 2 + 3) ou (6, 5). Por convenção assume-se que toda transição habilitada a partir da marcação inicial é disparada em $t = -\infty$. Assim, o ponto (6, 5) da figura 2.7 indica que 6 disparos estavam habilitados desde $t = -\infty$ e os outros disparos foram habilitados a partir de t = 5.

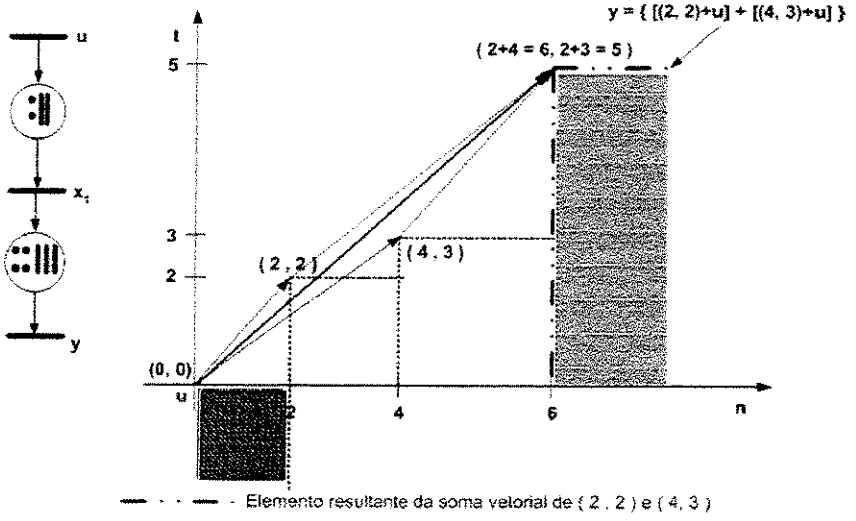


figura 2.7

Passagem da informação por lugares em série do GET e correspondencia no plano n x t

Na figura 2.8 abaixo pode-se observar a situação em que as informações chegam oriundas de duas entradas em "paralelo" com a transição y. também neste caso supõe-se que os disparos nas transições u_1 e u_2 correspondem ao cone com vértice no ponto (0, 0). No plano n×t a informação correspondente à transição y é dada pela união das regiões hachuradas cujo vértice é dado pelos pontos (2, 2) e (4, 4). Observa-se que a informação representada pela união dos cones (2, 2) e (4, 4 não pode ser simplificada contrariamente ao que aconteceria por exemplo com os cones dos pontos (2, 2) e (2, 1) (o cone de (2, 1) está contido no cone de (2, 2)).

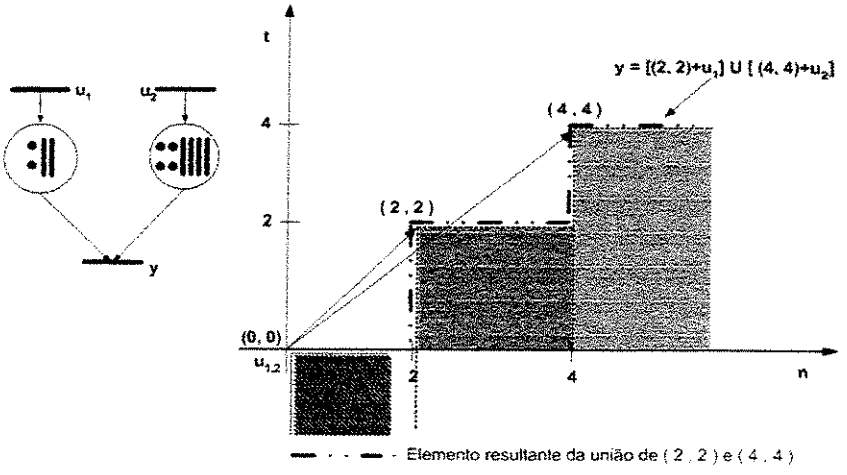


figura 2.8

Passagem da informação por lugares em paralelo do GET e correspondencia no plano n x t

Na figura 2.9 pode-se ver um grafo onde há lugares em "série" e em "paralelo" sendo a informação de y no plano n×t dada pela região hachurada (assumindo novamente que a transição u corresponde ao ponto (0, 0)).

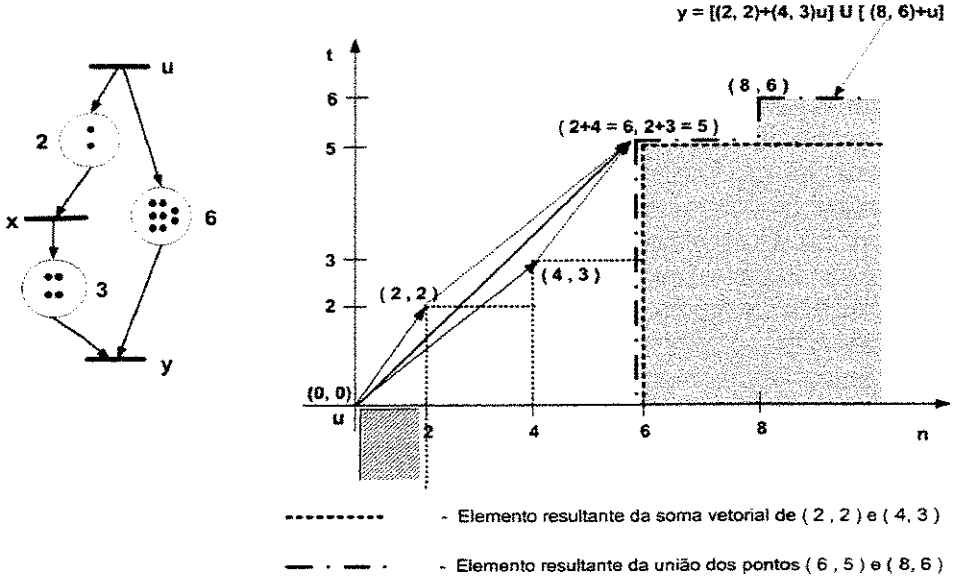


figura 2.9

Passagem da informação por lugares em série e em paralelo do GET e correspondencia no plano n x t

Se para representar a soma vetorial for utilizado o operador ⊗ e o operador ⊕ para representar a união ter-se-iam as duas situações de análise das informações contidas nos lugares do grafo conforme visto nos exemplos anteriores. Podemos utilizar essa notação para analisar o grafo da figura 2.9. Para a transição x temos que o único arco de "entrada" em x vem de um lugar cujo único arco de "entrada vem da transição u, ficando, nessa notação, a informação que chega a x escrita da seguinte forma: x = (2, 2) ⊗ u indicando simplesmente que x recebe informações de de um lugar do grafo ligado por um arco à transição u e nesse lugar há duas barras e duas fichas. Já a transição y pode ser escrita como: y = [(4, 3) ⊗ x] ⊕ [(8, 6) ⊗ u] pois tem-se informações vindas de dois arcos em "paralelo" vindos de lugares que recebem informações de x com 3 barras e 4 fichas e de u com 6 barras e 8 fichas respectivamente. Como a expressão para x já foi encontrada pode-se encontrar, por substituição, a expressão para x no lado esquerdo da união em y que fica: y = [(4, 3) ⊗ (2, 2) ⊗ u] ⊕ [(8, 6) ⊗ u] ou y = [(6, 5) ⊗ u] ⊕ [(8, 6) ⊗ u]. Pode-se notar que a saída é dada por uma expressão que, em termos de pontos no plano n×t pode ser vista como: (6, 5) ⊕ (8, 6) (pois u = (0, 0)) que podem ser vistas na figura 2.8 mostrando que pela manipulação das expressões chega-se a uma expressão que corresponde a saída da transição y vista na figura 2.9.

Considerando-se agora o conjunto de todas as trajetórias possíveis para uma transição pode-se definir duas operações entre trajetórias correspondendo às operações descritas anteriormente. Em resumo, Definindo-se os operadores binários:

$\oplus$ = União entre cones.

$\otimes$ = Deslocamento de vértices de cones de uma distância no plano igual aos vértices do outro operando (soma vetorial).

Pode-se reescrever as expressões anteriores como:

$x = (2, 2) \otimes u$

$y = [(4, 3) \otimes x] \oplus [(8, 6) \otimes u]$

$y = [(4, 3) \otimes (2, 2) \otimes u] \oplus [(8, 6) \otimes u]$

$y = [(6, 5) \otimes u] \oplus [(8, 6) \otimes u] = [(6, 5) \oplus (8, 6)] u$

O operador $\oplus$, definido como a união dos cones com vértice nos pontos da trajetória pode ser implementado pelas seguintes regras:

1. $(i, j) \oplus (i', j) = [\min(i, i'), j]$ (veja figura 2.10 abaixo)
2. $(i, j) \oplus (i, j') = [i, \max(j, j')]$ (veja figura 2.11 abaixo)

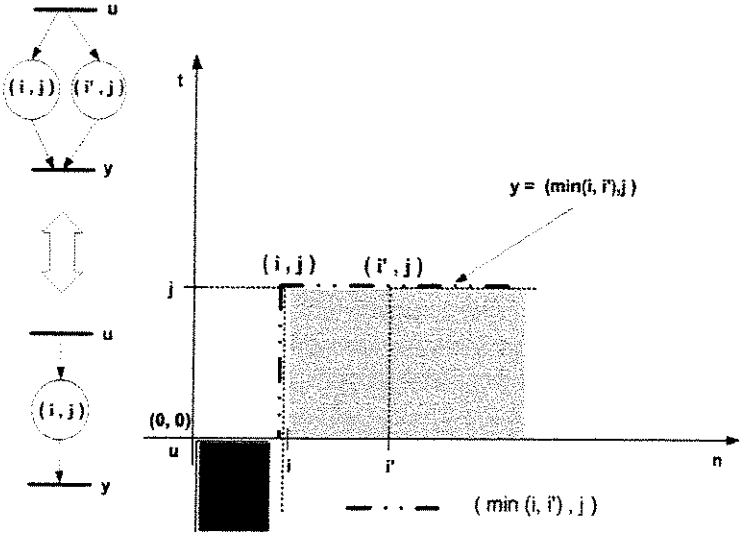


figura 2.10

visualização da regra 1 no plano n x t e a correspondente situação no GET

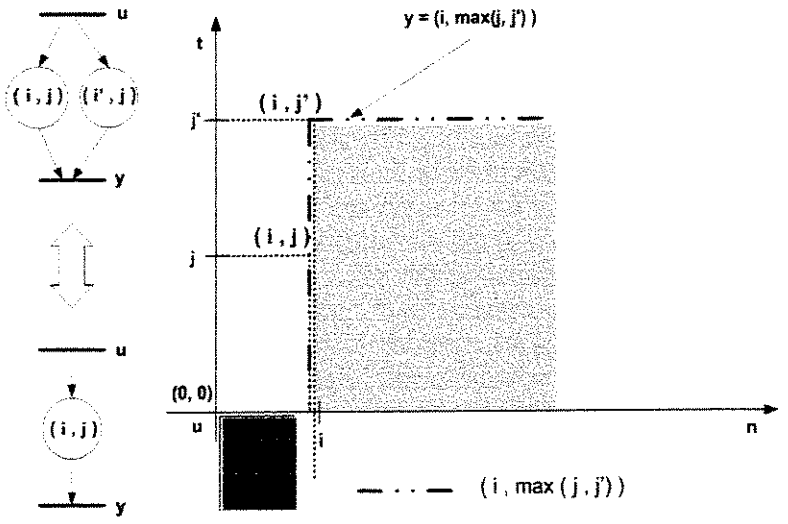


figura 2.11

visualização da regra 2 no plano n x t e a correspondente situação no GET

Utilizando esta notação pode-se agora analisar um grafo de maior complexidade que o exemplo da figura 2.4. No caso, o grafo da figura 2.4 cujas expressões para as transições podem ser dadas por:

1. $x_1 = (2, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 1) \otimes u$
2. $x_2 = (1, 1) \otimes x_1 \oplus (0, 2) \otimes u$
3. $y = (0, 0) \otimes x_1 \oplus (0, 0) \otimes x_2$

Substituindo 1 e 2 em 3 tem-se: $y = (0, 0) \otimes [(2, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 1) \otimes u] \oplus (0, 0) \otimes [(1, 1) \otimes x_1 \oplus (0, 2) \otimes u]$ daí, substituindo e aplicando as regras para a união dos cones a sudeste tem-se que:

$y = [(2, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 1) \otimes u] \oplus [(1, 1) \otimes x_1 \oplus (0, 2) \otimes u]$

$y = [(1, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 2) \otimes u]$

$y = (1, 1) \otimes [(2, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 1) \otimes u] \oplus (1, 1) \otimes [(1, 1) \otimes x_1 \oplus (0, 2) \otimes u] \oplus (0, 2) \otimes u$

$y = [(3, 2) \otimes x_1 \oplus (2, 2) \otimes x_2 \oplus (1, 2) \otimes u] \oplus [(2, 2) \otimes x_1 \oplus (1, 3) \otimes u \oplus (1, 3) \otimes u]$

$y = [(2, 2) \otimes x_1 \oplus (2, 2) \otimes x_2] \oplus (1, 3) \otimes u \oplus (0, 2) \otimes u$

$y = (2, 2) \otimes [(2, 1) \otimes x_1 \oplus (1, 1) \otimes x_2 \oplus (0, 1) \otimes u] \oplus (2, 2) \otimes [(1, 1) \otimes x_1 \oplus (0, 2) \otimes u] \oplus (1, 3) \otimes u \oplus (0, 2) \otimes u$

$y = [(4, 3) \otimes x_1 \oplus (3, 3) \otimes x_2 \oplus (2, 3) \otimes u] \oplus [(3, 3) \otimes x_1 \oplus (2, 4) \otimes u] \oplus (1, 3) \otimes u \oplus (0, 2) \otimes u$

$y = [(3, 3) \otimes x_1 \oplus (3, 3) \otimes x_2] \oplus (2, 4) \otimes u \oplus (1, 3) \otimes u \oplus (0, 2) \otimes u$

$$\vdots$$

$y = [(n, n) \otimes x_1 \oplus (n, n) \otimes x_2] \oplus (n-1, n+1) \otimes u \oplus (n-2, n) \otimes u \oplus (n-3, n-1) \otimes u \cdots \oplus (2, 4) \otimes u \oplus (1, 3) \otimes u \oplus (0, 2) \otimes u$

o que nos dá a seqüência de pontos (0,2), (1,3), (2,4), ... do plano n×t que deslocados de u marcam os vértices dos "cones" que juntos formam a curva que representa a trajetória da transição de saída y. Conforme se pode ver na figura 2.12 abaixo:
Nota: A transição u é a entrada impulsiva da rede e a ela será associada o ponto (0, 0) do plano n×t.

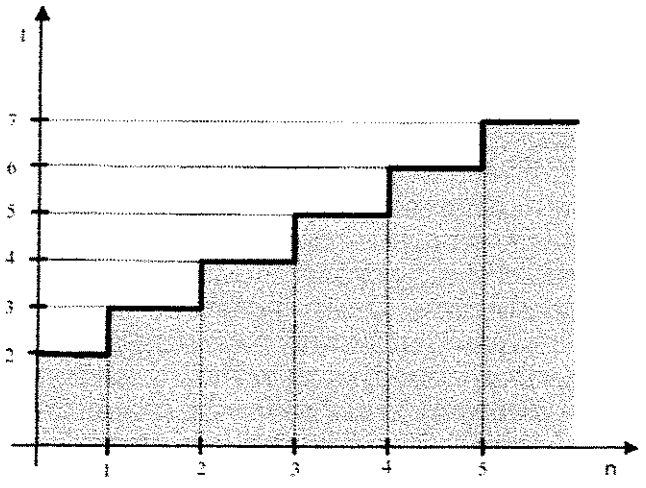


figura 2.12
Trajetória de saída do grafo da digura 1.4

Portanto, pode-se representar (de modo insuficiente, pois é necessário o cone a sudeste com vértice nestes pontos para uma representação completa) a informação correspondente a uma transição de um grafo a eventos pela coleção de pontos no plano $\mathbb{Z}^2$ onde $\{(n_i, t_i) : i \in I \subset \mathbb{N}\}$. Visando uma representação algébrica, tais coleções de pontos podem ser associadas a polinômios do tipo:

$\sum_{i \in I} \gamma^{n_i} \delta^{t_i}$ onde associaremos γ ao índice n_i (fichas) e δ ao índice t_i (barras) do par de pontos.

A motivação para tal representação pode ser explicada utilizando-se os dois grafos da figura 2.13 onde no grafo da esquerda, se a informação que chega a t_1 é dada por : $t_1 = \sum \gamma^i \delta^j$, então $t_2 = \gamma \otimes \sum \gamma^i \delta^j$ sendo que o operador $\otimes$ mantém o seu comportamento conforme descrito anteriormente resultando então em $t_2 = \sum \gamma^{i+1} \delta^j$. Já no grafo da direita, se a informação que chega a t_1 é dada por : $t_1 = \sum \gamma^i \delta^j$, então $t_2 = \delta \otimes \sum \gamma^i \delta^j$ sendo que o operador $\otimes$ mantém também, neste caso, o seu comportamento conforme descrito anteriormente resultando então em $t_2 = \sum \gamma^i \delta^{j+1}$.

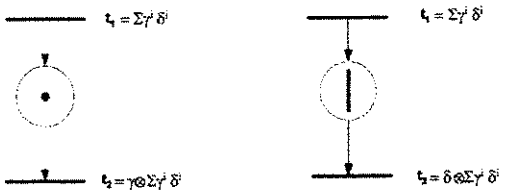


figura 2.13

Representação da passagem de informação pelos lugares do grafo por polinômios

Utililizando-se a representação acima a figura 2.7 pode ser vista como na figura 2.14 abaixo:

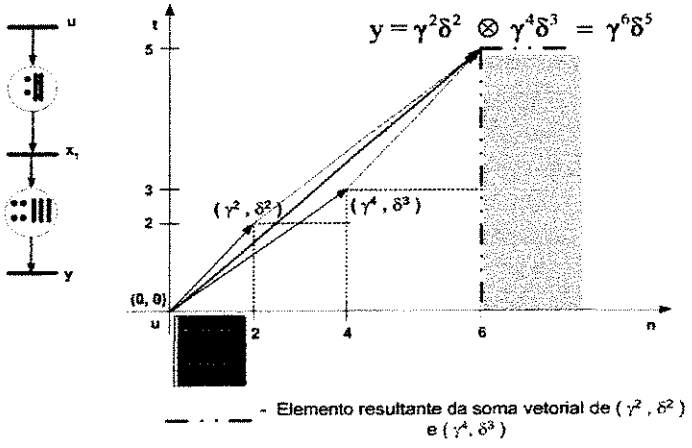


figura 2.14

Soma vetorial de polinômios do tipo $\gamma^{n_i}\delta^{t_i}$ e a correspondente situação no GET

E a figura 2.8 como:

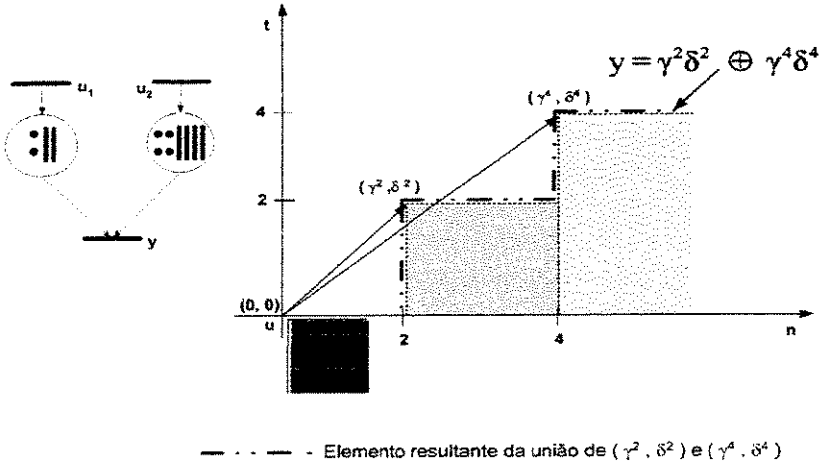


figura 2.15

União de polinômios do tipo $\gamma^{n_i}\delta^{t_i}$ e a correspondente situação no GET

Conforme foi dito antes a representação da trajetória de saída de uma transição é incompleta se apresentada apenas pelos pontos no plano n×t, sendo necessários os cones a sudeste com vértice nestes mesmos pontos do plano n×t. Posteriormente serão acrescentadas regras para tornar possível que a soma e o produto entre cones a sudeste produzam também cones a sudeste no plano n×t.

A partir de agora os conceitos utilizados para representar a trajetória de uma transição no plano n×t apresentados de modo intuitivo serão formalizados numa estrutura algébrica chamada de dióide onde os elementos de um dióide são as regiões do plano n×t que se encontram a sudeste da trajetória de uma transição e as operações de soma e produto do dióide são definidas de modo a refletir as relações entre trajetórias diferentes do grafo a eventos se comportando conforme foi discutido nos exemplos anteriores. Também serão apresentadas definições como série de potência, polinômios, congruência entre outras preparando as definições de dióide L e dióide *M*, mostrando-se finalmente que um grafo a eventos pode ser modelado por um sistema de equações algébricas no dióide *M*. A estrutura algébrica utilizada na literatura (Bacelli,. et al, 1992) para modelagem de GET são os **semi - anéis idempotentes**, também denominados dióides, definidos a seguir.

Definição 2.7
Dióide.

Um conjunto D munido de duas operações internas, soma ($\oplus$) e produto ($\otimes$) é um dióide se os 6 seguintes axiomas forem verificados:

Axioma 1: **Associatividade.**

$\forall a,b,c \in D$

$(a \oplus b) \oplus c = a \oplus (b \oplus c)$

$(a \otimes b) \otimes c = a \otimes (b \otimes c)$

Axioma 2: **Comutatividade da adição.**

$\forall a,b \in D \quad a \oplus b = b \oplus a$

Axioma 3: **Distributividade.**

$\forall a,b,c \in D$

$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$

$c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$

Axioma 4: **Elemento nulo e elemento identidade.**

$\exists \varepsilon \in D : \forall a \in D,\ a \oplus \varepsilon = a$

$\exists e \in D : \forall a \in D,\ a \otimes e = a$

Axioma 5: **Absorção pelo elemento nulo.**

$\forall a \in D,\ a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$

Axioma 6: **Idempotência da adição**

$\forall a \in D,\ a \oplus a = a$

Pode-se citar como exemplos de dióides:

D	$\oplus$	$\otimes$	
$\mathbb{R} \cup \{+\infty\}$	max	+	
$\mathbb{R} \cup \{-\infty\}$	min	+	
$\{0,1\}$	max	min	← dióide $\mathbb{B}$
$\wp(\mathbb{R}^2)$	$\cup$	+ (soma vetorial)	

Tabela 2.1

Exemplos de dióides

Ressalta-se como peculiaridades dos dióides a ausência de um elemento simétrico aditivo bem como a idempotência da adição (axioma 6).

Definição 2.8
Ordenamento em Dióides

A seguinte relação estabelece um ordenamento parcial num dióide $a \geq b$ se e somente se $a \oplus b = a$. Δ

De modo geral o produto é isotônico, ou seja, se: $a \geq b$ então $a \otimes c \geq b \otimes c$.

Uma propriedade interessante dos dióides é o fato de que a partir de um dióide é possível definir uma família de dióides, a partir das seguintes definições:

Definição 2.9
Dióide Comutativo

Um dióide é comutativo sempre que a multiplicação for comutativa. △

Definição 2.10
Série Formal de Potência

Uma série formal de potência em p variáveis (comutativas) $z_1 \ldots z_p$ com coeficientes num dióide D é uma aplicação f de $\mathbb{Z}^p$ ou $\mathbb{N}^p$ em D, de modo que para todo $k = (k_1, k_2, \ldots k_p) \in \mathbb{Z}^p$ ou $\mathbb{N}^p$, f(k) representa os coeficientes de $z_1^{k_1} \ldots z_p^{k_p}$. Pode-se denotar uma série de potências por:

$$f = \bigoplus_{k \in \mathbb{Z}^p \text{ ou } k \in \mathbb{N}^p} f(k_1,\ldots,k_p) z_1^{k_1} \ldots z_p^{k_p} \quad \triangle$$

Deve-se notar que, por exemplo, f(7) não representa o valor da série para $z = 7$, mas o coeficiente de z^7.

Definição 2.11
Soma e Produto de Séries Formais.

Considere o conjunto de todas as séries formais obtidas a partir de um dióide D e de p variáveis $z_1 \ldots z_p$. A soma e o produto de duas séries f_1 e f_2 pertencentes a este conjunto, são dadas por:

- $g = f_1 \oplus f_2$ se e somente se $\forall k \in \mathbb{N}^p$ ou $\mathbb{Z}^p$, $g(k) = f_1(k) \oplus f_2(k)$.
- $g = f_1 \otimes f_2$ se e somente se $\forall k \in \mathbb{N}^p$ ou $\mathbb{Z}^p$, $g(k) = \bigoplus_{k_1 + k_2 = k} (f_1(k_1) \otimes f_2(k_2))$.△

Teorema 2.1 (Bacelli,. et al, 1992)

O conjunto de séries formais obtido a partir de um dióide D e de p variáveis $z_1 \ldots z_p$, munido das operações soma e produto acima definidas é um dióide, (denotado por $D[\![z_1 \cdots z_p]\!]$).♦

Definição 2.12
Suporte, Grau e Valoração de uma série formal de Potências.

O **suporte** de uma série f em p variáveis é o conjunto $\text{supp}(f) = \{k \in \mathbb{Z}^p / f(k) \neq \epsilon\}$ onde ϵ representa o elemento nulo.△

É possível ordenar parcialmente o suporte de uma série de potências através de: $(k_1, k_2, \ldots k_p) \times (k_1', k_2', \ldots k_p')$ se e somente se $\forall i, k_i \leqslant k_i'$.

O **grau** de uma série formal de potências é definido como: $\text{grau}(f) = \sup\{\text{supp}(f)\}$. Onde o supremo e o ínfimo se refere ao ordenamento parcial definido para o suporte.△

A **valoração** de uma série formal de potências é definida como: val(a) = inf{supp(f)}. △

Definição 2.13
Polinômio

Um polinômio é uma série de potência finita, portanto com suporte finito, com expoentes em $\mathbb{N}$. △

Como exemplo, seja um polinômio bi-variável (γ e δ) definido sobre um dióide D dado por p = $a_1\gamma^2\delta^5 \oplus a_2\gamma^4\delta^1$ onde a_1 e $a_2 \in$ D. Tem-se supp(p) = {(2,5),(4,1)}, grau(p) = (4,5) e val(p) = (2,1).

Definição 2.14
Classes de Equivalência

Uma relação de equivalência R (isto é, uma relação reflexiva, simétrica e transitiva) definida sobre um dióide D, define classes de equivalência em D. A classe de equivalência que contém o elemento a $\in$ D será denotada por $[a]$ e o conjunto das classes de equivalência é chamado de quociente de D em relação a R. △

Definição 2.15
Congruência

Uma relação de equivalência (ou seja, reflexiva, simétrica e transitiva) definida num dióide D é chamada de **congruência** se e somente se for compatível com $\oplus$ e $\otimes$, isto é :

$\forall$ a,b,c $\in$ D : $a \equiv b \Rightarrow a \oplus c \equiv b \oplus c$

$\forall$ a,b,c $\in$ D : $a \equiv b \Rightarrow a \otimes c \equiv b \otimes c$ △

Teorema 2.2 (Cohen,. et al, 1989)

Dada uma congruência em D é possível definir as seguintes operações no quociente de D:

$[a] \oplus [b] = [a \oplus b]$ e $[a] \otimes [b] = [a \otimes b]$. Além disso, o conjunto quociente munido das duas operações anteriormente definidas é um dióide.▲

Nota: A possibilidade de definir tais operações advém do fato que: $a \equiv a'$ e $b \equiv b' \Rightarrow [a] \oplus [b] = \left[a' \oplus b\right]$ e $[a] \otimes [b] = \left[a' \otimes b\right]$.

Definição 2.16
Operação Estrela (*).

A operação estrela é definida como: $a^* = e \oplus a \oplus a^2 \oplus a^3 \oplus ...$, onde "e" representa o elemento unitário. △

Uma propriedade útil em dióides comutativos é:

1) $(a \oplus b)^* = (b \otimes a)^*$

Definição 2.17
Equivalência módulo z.

Seja D um dióide comutativo e sejam a, b, z ∈D, e. Diz-se que a e b são equivalentes módulo z(a ≡ b mod z) se e somente se $az^* = bz^*$. Esta relação é uma congruência, podendo-se notar por $[a]_z$ a classe de equivalência definida por a (todos os elementos de D equivalentes módulo z ao elemento a).△

Conforme o teorema 2.2 a relação de congruência "equivalência módulo z" define um dióide quociente em D. Este dióide será denotado por D/z.

2.5 Dióide L e Dióide *M*.

Nesta seção definir-se-á um dióide apropriado para a descrição da dinâmica de GET's, isto é, cujos elementos sejam trajetórias associadas a transições e cujas operações correspondem às operações envolvidas num GET (união de cones e soma vetorial).

Definição 2.18
Dióide L

Considere-se o dióide $\mathbb{B} = \langle \{0, 1\}, \max, \min \rangle$ (tabela 2.1).

Define-se L como o **conjunto de séries de potências** em **duas variáveis** γ e δ com **coeficientes** em $\mathbb{B}$ e **expoentes** em $\mathbb{Z}$, isto é $L = \mathbb{B}[\![\gamma, \delta]\!]$.△

De acordo com o teorema 2.1 L é um dióide.

O dióide L não é adequado para representar a informação na transição de um GET de modo conveniente, conforme mostrado anteriormente.

Como foi visto, a informação referente às transições pode ser insuficientemente representada através de uma coleção de pontos $\{(n_i, t_i) / i \in I \subseteq \mathbb{N}\}$ no plano, os "cones a sudeste" de cada ponto do plano (n x t) precisam ser acrescentados à representação (figura 2.16). Pode-se representar uma coleção de pontos por polinômios do tipo $\sum_{i \in I} \gamma^{n_i} \delta^{t_i}$.

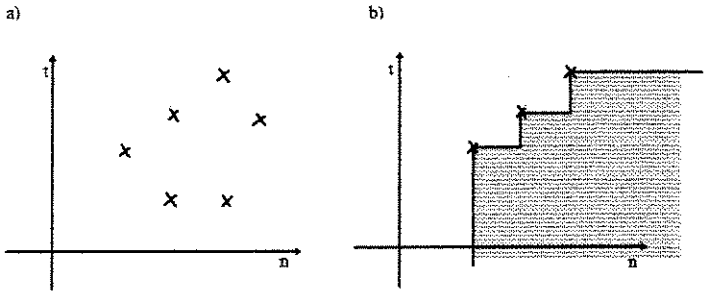


figura 2.16

Pontos do plano n x t e cones a sudeste com vértices nos pontos do plano n x t.

O conjunto destes polinômios é igual ao dióide $L = \mathbb{B}[\![\gamma, \delta]\!]$ onde $\mathbb{B}$ é o dióide booleano definido na tabela 2.1.

Os dióide L não é ainda a representação formal desejada pois seus elementos constituem pontos de $\mathbb{Z}^2$ e não "cones a sudeste". É necessário, portanto, criar classes de equivalência em L constituídas por todos os pontos a sudeste de um ponto dado. A seguinte congruência define as classes de equivalência desejadas:

$\gamma^m \delta^d \equiv \gamma^n \delta^t$ se e somente se $\gamma^m \delta^d \equiv \gamma^n \delta^t \otimes \left(\gamma \oplus \delta^{-1}\right)^*$

De fato, tem-se que:

$\gamma^n \delta^t \otimes \left(\gamma \oplus \delta^{-1}\right)^* = \gamma^n \delta^t \otimes \gamma^* \otimes (\delta^{-1})^*$.(pela equação 1) Expandindo-se as operações estrela tem-se:

$\gamma^n \delta^t \otimes \gamma^* \otimes (\delta^{-1})^* = \gamma^n \delta^t \otimes \left(e \oplus \gamma \oplus \gamma^2 \oplus \gamma^3 \oplus \ldots\right) \otimes (e \oplus \delta^{-1} \oplus \delta^{-2} \oplus \delta^{-3} \oplus \ldots)$.

Os dois multiplicandos à direita do monômio podem ser vistos nas figuras 2.17 e 2.18 abaixo:

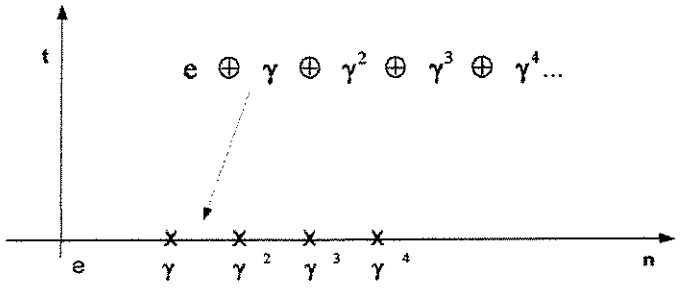


figura 2.17

Polinômio $(e \oplus \gamma \oplus \gamma^2 \ldots)$ visto no plano n x t

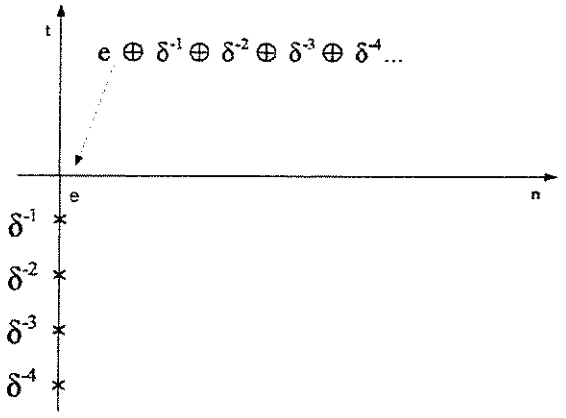


figura 2.18

Polinômio $(e \oplus \delta^{-1} \oplus \delta^{-2} \ldots)$ visto no plano n x t

Pode-se ver na figura 2.19 abaixo a representação no plano n×t de um monômio do tipo $\gamma^n \delta^t$ e também do produto $\gamma^n \delta^t (\gamma \oplus \delta^{-1})^*$. Define-se portanto o dióide M como L módulo $(\gamma \oplus \delta^{-1})^*$ (o quociente de L por $(\gamma \oplus \delta^{-1})^*$).

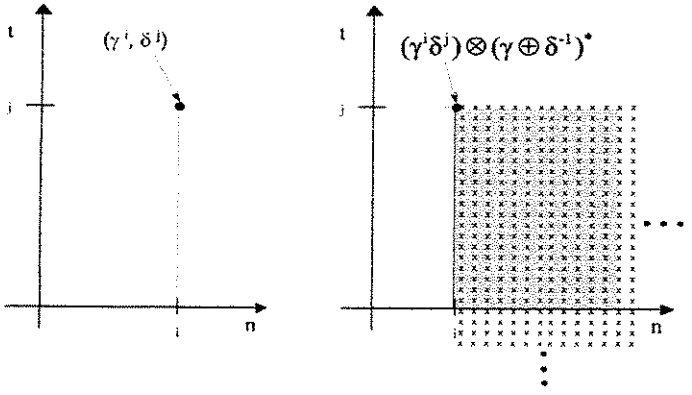


figura 2.19

Monômio $\gamma^n \delta^t$ e o produto $\gamma^n \delta^t (\gamma \oplus \delta^{-1})^*$ no plano n x t

É imediato observar que a soma de elementos do dióide M corresponde a uma união de "cones a sudeste" e que o produto corresponde a um deslocamento no plano n×t. O dióide M é portanto adequado para a modelagem das informações relativas às transições em um GET, conforme discutido na seção 2.4. Na próxima seção discutir-se-á esta modelagem.

Equações para o GET no Dióide *M*.

Uma vez que o dióide M é capaz de representar os cones a sudeste dos pontos no plano n×t pode-se representar as trajetórias assim como as operações entre trajetórias de cada transição do GET

através de equações lineares escritas no dióide M permitindo descrever o comportamento, no plano n×t, da trajetória da(s) saída(s) do GET.

Para modelar um GET, associa-se a cada uma de suas transições uma variável que assume valores em M. As transições de entrada, isto é, aquelas disparadas externamente ($I(t) = \varnothing$) serão associadas as variáveis $u_1,...,u_m$. Às transições de saída (com $o(t) = \varnothing$) serão associadas as variáveis $y_1,...,y_p$. Às transições internas (todas as outras) serão associadas as variáveis $x_1,...,x_n$. Das discussões da seção 2.4 concluiu-se que a situação representada na figura 2.20 pode ser modelada por: $x_k = \gamma^{m_i}\delta^{d_i}x_i \oplus \gamma^{m_j}\delta^{d_j}x_j$ onde m_i e m_j representam as marcações iniciais e d_i e d_j representam os atrasos dos lugares da rede. portanto, um grafo de eventos pode ser modelado pelas seguintes equações algébricas lineares $x = Ax \oplus Bu$ e $y = Cx$ onde os vetores u, x e y de dimensões m, n e p correspondem a m transições de entrada, n transições internas e p transições de saída do grafo, respectivamente e as matrizes $A^{n\,x\,n}$, $B^{n\,x\,m}$ e $C^{p\,x\,n}$ são descritas no dióide apropriado com elementos polinomiais do tipo série de potência em (γ, δ) e coeficientes em $\{\epsilon, e\}$ onde ϵ = elemento nulo = $\{\gamma^{\infty}, \delta^{-\infty}\}$ e e = elemento unitário = $\{\gamma^0, \delta^0\}$. Na figura 2.21 tem-se a representação matricial das equações lineares para um GET discriminando-se as matrizes A, B e C, bem como os vetores u, x e y.

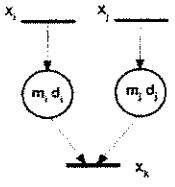


figura 2.20

Esquema de um GET

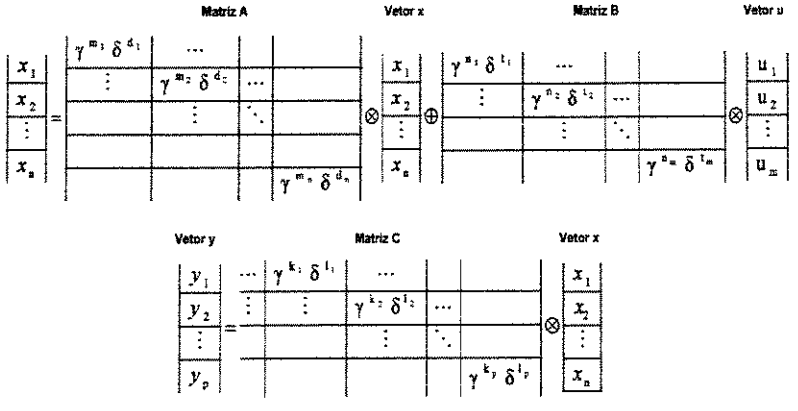


figura 2.21

Equações lineares na forma matricial para o GET genérico da figura

O conjunto Θ de equações de um grafo a eventos escritas no dióide associado é da forma: $x_i = \bigoplus_{j=1}^{n} \gamma^{m_{ij}} \delta^{d_{ij}} x_j$ onde: m_{ij} e d_{ij} são a marcação inicial e o atraso do lugar com transição de entrada x_j e transição de saída x_i.

A saída do sistema é encontrada resolvendo-se as equações $x = Ax \oplus Bu$ que, por Cohen (1989) teorema 9-ii possui solução mínima dada por $x = A^*B$ e $y = Cx$, que substituindo o valor de x na equação de y dá:

$$\left\{ \begin{array}{l} x = A^*Bu \\ y = CA^*Bu \end{array} \right\}$$

Sendo $H = CA^*B$ chamada de **função de transferência** do sistema. A matriz A^* pode ser calculada recursivamente (Baccelli et al., 1992). Seja $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ uma partição de A tal que d seja escalar. Então: $A^* = \begin{bmatrix} a^* \oplus d^*a^*bc(a \oplus bc)^* & d^*(a \oplus bc)^*b \\ d^*c(a \oplus bc) & d^* \oplus d^*c(a \oplus bc)^*b \end{bmatrix}$

A **matriz de transferência** do sistema modelado através de grafos a eventos no dióide *M* é definida por $H = CA^*B$ tendo sua interpretação similar ao da teoria de controle onde cada elemento h_{ij} corresponde à saída y_i quando $u_j = e$, $u_k = \epsilon$, $\forall k \neq j$, com $e = \gamma^0\delta^0$ e $\epsilon = (+\infty, -\infty)$. Tem-se que que a entrada $u_j = e$ pode ser vista como "impulso" ou seja, em $t = 0$ há um número infinito de fichas já a entrada $u_k = \epsilon$ pode ser vista como $u_k = \gamma^{\infty}\delta^{-\infty}$ correspondendo a um número infinito de fichas em $t = -\infty$ devendo ser interpretada como a entrada menos restritiva possível e, quando aplicada resulta em $= \epsilon$, ou seja, qualquer saída é possível.

Nota: Supõe-se que a marcação inicial do grafo está disponível desde um tempo $t = -\infty$.

Usando a rede da figura 4 como exemplo temos que:

$x_1 = \gamma^2\delta x_1 \otimes \gamma\delta x_2 \oplus \delta u$

$x_2 = \gamma\delta x_1 \otimes \epsilon x_2 \oplus \delta^2 u$

$y = x_1 \oplus x_2$

$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \gamma^2\delta & \gamma\delta \\ \gamma\delta & \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \oplus \begin{pmatrix} \delta \\ \delta^2 \end{pmatrix} u$ e, para a saída "y" tem-se: $y = \delta^2(\gamma\delta)^*u$

(veja figura 2.22 a seguir).

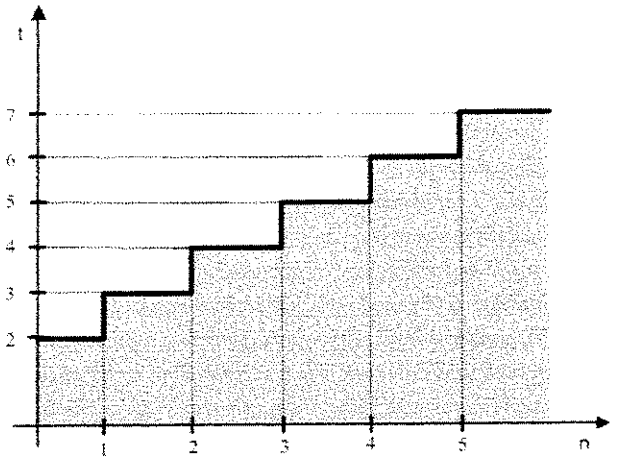


figura 2.22

Trajetória de saída $\delta^2(\gamma\delta)^* u$ para o GET da figura 2.4

Usando a rede abaixo pode-se visualizar um exemplo de uma rede com duas entradas u_1 e u_2, mostrando que a saída pode ser vista como a soma (no dióide) da saída em função de u_1 e u_2 respectivamente.

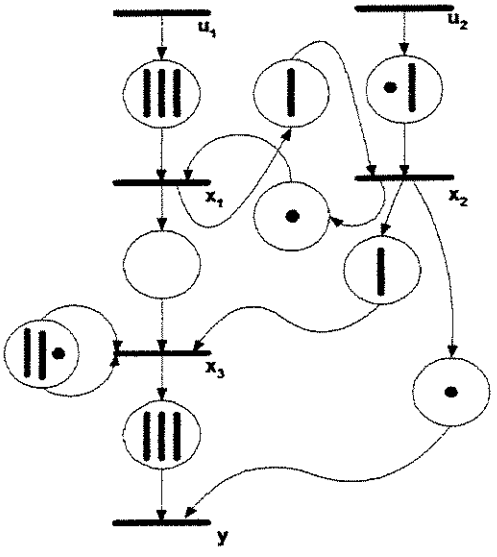


figura 2.23

GET com duas entradas u_1 e u_2

Tem-se as seguintes equações:

$$x_1 = (\epsilon x_1 \oplus \gamma x_2 \oplus \epsilon x_3) \oplus (\delta^2 u_1 \oplus \epsilon u_2)$$

$x_2 = (\delta x_1 \oplus \epsilon x_2 \oplus \epsilon x_3) \oplus (\epsilon u_1 \oplus \gamma\delta u_2)$

$x_3 = (ex_1 \oplus \delta x_2 \oplus \gamma\delta^2 x_3) \oplus (\epsilon u_1 \oplus \epsilon u_2)$

$y = \epsilon x_1 \oplus \gamma x_2 \oplus \delta^3 x_3$

Na forma matricial fica:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \epsilon & \gamma & \epsilon \\ \delta & \epsilon & \epsilon \\ e & \delta & \gamma\delta^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \oplus \begin{bmatrix} \delta^2 & \epsilon \\ \epsilon & \gamma\delta \\ \epsilon & \epsilon \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

ficando a saída y como:

$$y = \begin{bmatrix} \epsilon & \gamma & \delta^3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

A matriz de transferência é dada por:

$y = \delta^8 (\gamma\delta^2)^* u_1 \oplus \gamma\delta^5 (\gamma\delta^2)^* u_2$

que pode ser vista graficamente como:

saída em função de u_1 ($u_1 = e$ e $u_2 = \epsilon$) $\delta^8 u_1 \oplus \gamma\delta^{10} u_1 \oplus \gamma^2\delta^{12} u_1 \oplus \gamma^3\delta^{14} u_1 \oplus \ldots$ (figura 2.24):

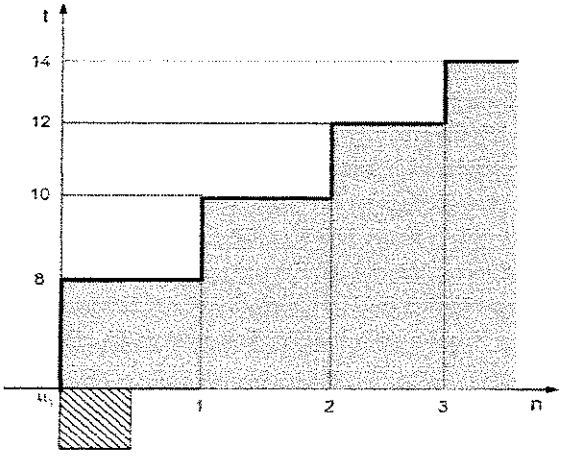


figura 2.24

Saída do GET da figura 2.13 em função da transição de entrada u_1

saída em função de u_2 ($u_1 = \epsilon$ e $u_2 = e$) $\gamma\delta^5 u_2 \oplus \gamma^2\delta^7 u_2 \oplus \gamma^3\delta^9 u_2 \oplus \gamma^4\delta^{11} u_2 \oplus \ldots$ (figura 2.25):

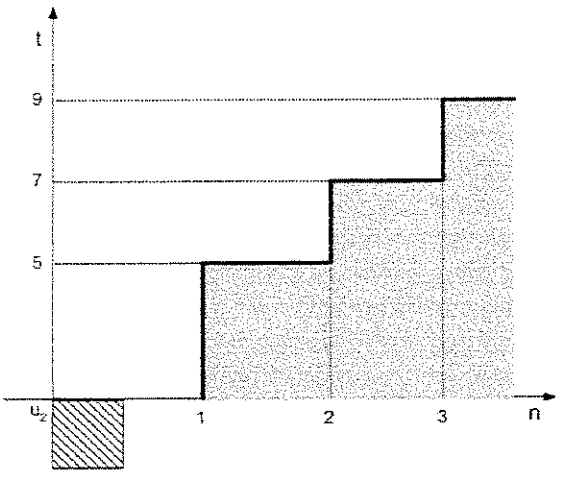


figura 2.25

Saída do GET da figura 2.13 em função da transição de entrada u_2

2.6 Conclusão

Neste capítulo foi apresentada uma noção das redes de Petri, particularmente de uma classe de redes de Petri chamada de grafo de eventos, onde os grafos de eventos temporizados foram analisados. Posteriormente a representação algébrica dos grafos de eventos foi mostrada assim como a obtenção, via álgebra, da "saída" de um grafo de eventos a partir de suas entradas. Generalizou-se esta álgebra numa estrutura denominada dióide apresentando-se algumas de suas características formais, a representação do grafo de eventos e um método de obtenção do comportamento do grafo de eventos através de sua matriz de transferência. No próximo capítulo sera discutida a lógica modal NK e sua equivalência com o dióide assim como sua ferramenta de prova lógica tableau que pode ser utilizada para verificação de especificações.

CAPÍTULO 3

Lógica NK e Dióides.

3.1 Introdução.

Neste capítulo será feita uma apresentação da lógica NK (*Magossi*, 1998) e sua correlação com a álgebra de dióides. O mecanismo de prova tableau tradicional (*Smullyan*, 1968) será discutido, assim como o tableaux NK, (o mecanismo de prova da lógica NK) e sua utilização para a verificação de especificações.

A lógica NK é do tipo modal (Graham, 2001 e Hughes e Cresswell, 1996). Uma lógica proposicional modal pode ser vista como uma extensão da lógica proposicional clássica e possui como característica a presença de operadores não funcionais veritativos a que chamaremos operadores modais. A semântica das lógicas modais proposicionais é definida a partir do conceito de "mundo possível". Um "mundo possível" pode ser visto intuitivamente como uma atribuição de valor-verdade a todas as variáveis proposicionais da lógica. Nele, os operadores da lógica clássica atuam como usualmente, podendo ser definidos através de uma tabela-verdade. Para definir completamente uma interpretação para uma lógica modal é necessário definir um conjunto de mundos possíveis e uma relação binária (R) entre eles. Para cada mundo possível é feita uma atribuição de valores-verdade e por conseqüência o valor-verdade das fórmulas que se utilizam exclusivamente de operadores clássicos também é determinado. A determinação do valor verdade de uma fórmula modal num dado mundo possível é feita a partir do valor-verdade do operando nos mundos que estão em relação com o mundo dado. Um exemplo tradicional é o operador modal L cuja semântica é "necessário". Diz-se que LX é verdadeira num mundo possível w se e somente se X é verdadeira em todos os mundos w' tais que wRw'.

A lógica NK, tem por objetivo modelar sistemas descritíveis por equações lineares no dióide *M* Portanto, a lógica NK permite a modelagem de sistemas descritos por GET's, através de uma representação equivalente à representação via dióide *M* utilizando-se como mecanismo de verificação de especificações o tableau semântico com o seguinte procedimento: Parte-se de um GET e uma especificação que se deseja verificar, descrevendo-se a especificação e as relações entre as transições do GET através de fórmulas da lógica NK. A especificação será satisfeita se e somente se a fórmula que a expressa for conseqüência lógica do conjunto de fórmulas que descreve o GET, o que pode ser

verificado através do tableau analítico. Desse modo transforma-se o problema de verificação de especificações em um GET ao problema de determinar se uma fórmula da lógica NK é conseqüência lógica de um conjunto de fórmulas.

3.2 A Linguagem da Lógica Modal NK.

Algumas definições importantes relativas à linguagem, ou **sintaxe**, da lógica NK são apresentadas abaixo:

Definição 3.1
Alfabeto.

O alfabeto usado pela lógica NK será constituído de:

- Conectivos : $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, G, D.
- Variáveis proposicionais: a, b, c,...,z, ... a_1, b_1, c_1,...,z_1, ..., a_2, b_2, c_2,...,z_2 ...
- Sinais de pontuação: (,). $\triangle$

Definição 3.2
Fórmulas.

Uma fórmula é uma seqüência finita de símbolos do alfabeto satisfazendo às seguintes condições recursivas:

- Toda variável proposicional é uma fórmula (chamada de fórmula atômica).
- Se X é fórmula, então $\neg$X também o será.
- Se X e Y são fórmulas então (X $\wedge$ Y), (X $\vee$ Y), (X $\rightarrow$ Y), (X $\leftrightarrow$ Y) também são fórmulas.
- Se X é fórmula então GX, DX, também o são.
- Só são fórmulas as expressões advindas das quatro regras anteriores. $\triangle$

Definição 3.3
Subfórmulas.

X é uma subfórmula de Y se:

- X = Y.

- $Y = (Y1 \wedge Y2)$, ou $(Y1 \vee Y2)$ ou $(Y1 \rightarrow Y2)$ ou $(Y1 \leftrightarrow Y2)$, onde X é uma subfórmula de Y1 ou Y2 .
- $Y = \neg Y1$ ou GY1 ou DY1, sendo X uma subfórmula de Y1. $\triangle$

Nota: Variáveis não tem subfórmulas próprias.

Definição 3.4
Grau de uma Fórmula.

O grau de uma fórmula é o número de ocorrências dos símbolos $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, G e D. Daí:

- Variáveis proposicionais são de grau zero.
- Se X é de grau n então $\neg X$, GX e DX são de grau (n+1).
- Se X e Y são de graus n e m, então $(X \wedge Y)$, ou $(X \vee Y)$ ou $(X \rightarrow Y)$ ou $(X \leftrightarrow Y)$ são de grau (n+m+1). $\triangle$

Definição 3.5
Modalidades.

Define-se modalidade como uma seqüência finita de operadores G e D, podendo-se escrever ϕ^n X para indicar que a fórmula X está sujeita a n iterações do operador ϕ (ϕ = G ou ϕ = D), com n = 1,2,3... .

Assim, para qualquer operador ϕ:

- $\phi^0 X = X$.
- $\phi^n X = \phi\phi^{n-1} X$, para $n > 0$. $\triangle$

Exemplo: G^4D^5X corresponde a 4 iterações do operador G e 5 iterações do operador D.

3.3 Modelos Proposicionais NK.

Segue-se uma apresentação de modelos baseados na **semântica** de mundos possíveis para a lógica modal proposicional NK.

Definição 3.6
Mundos Possíveis.

Seja W um conjunto não vazio e enumerável, seus elementos serão chamados de mundos possíveis

e notados pela letra w (w',w'',w'''...). △

Definição 3.7
Propriedades de uma Relação Binária entre os Elementos de W.

De modo geral adota-se a seguinte notação. Se K é uma relação binária entre os elementos de W então:

- $w\,K^0\,w'$ se e somente se $w = w'$.
- $w\,K^n\,w'$ se e somente se existe $w'' \in W$ onde $w\,K\,w''$ e $w''\,K^{n-1}\,w'$, com $n > 0$. △

Definição 3.8
NK-Estrutura.

Seja W um conjunto de mundos possíveis e sejam R e S duas relações binárias entre elementos de W. O terno $\langle W, R, S \rangle$ será chamado de NK-estrutura, se existir uma bijeção $f: W \rightarrow \mathbb{Z}^2$ tal que $\forall$ w', w'' ∈ W, e sendo f(w')=(i, j) tem-se

$$\{w'\,R\,w'' \iff f(w'') = (i-1, j)$$

e

$$\{w'\,S\,w'' \iff f(w'') = (i, j-1) \quad \triangle$$

Observa-se que esta definição de NK-estrutura constitui uma formalização ligeiramente diferente daquela proposta originalmente por Magossi(1998). Entretanto, para fins da correspondência com o dióide M discutida na seção 3.5, as duas formalizações são equivalentes.

Definição 3.9
Valoração.

Seja F o conjunto de fórmulas da lógica NK e $\langle W, R, S \rangle$ uma NK-estrutura. Sejam X, Y ∈ F, w ∈ W e p uma variável proposicional. Uma **valoração** é uma função $v: F \times W \rightarrow \{t, f\}$, com v satisfazendo as seguintes condições.

- $v(p, w) = t$ ou (exclusivamente) $v(p, w) = f$.
- $v((\neg X), w) = t$ se e somente se $v(X, w) = f$.
- $v((X \wedge Y), w) = t$, se e somente se $v(X, w) = t$ e $v(Y, w) = t$.
- $v((X \vee Y), w) = t$, se e somente se $v(X, w) = t$ ou $v(Y, w) = t$.

- $\nu((X \rightarrow Y), w) = t$, se e somente se $\nu(X, w) = f$ ou $\nu(Y, w) = t$.
- $\nu((X \leftrightarrow Y), w) = t$, se e somente se $\nu(X, w) = t$ e $\nu(Y, w) = t$.
- $\nu(DX, w) = t$, se e somente se $\exists w' \in W$ tal que $w S w'$ e $\nu(X, w') = t$.
- $\nu(GX, w) = t$, se e somente se $\exists w' \in W$ tal que $w R w'$ e $\nu(X, w') = t$. Δ

Definição 3.10
Modelo Proposicional Modal NK.

Um modelo proposicional modal para a lógica NK é uma quadra $<W, R, S, \nu>$ onde $<W, R, S>$ é uma NK-estrutura e ν é uma valoração.Δ.

Definição 3.11
Fórmula Verdadeira.

Diz-se que uma fórmula X é **verdadeira** no modelo $<W, R, S, \nu>$ se, para todo $w \in W$ em $<W, R, S, \nu>$ tivermos $\nu(X, w) = t$. Δ

Definição 3.12
Fórmula Satisfatível

Uma fórmula X é dita **satisfatível** se e somente se existe uma valoração ν, tal onde $\nu(X) = t$ (X é verdadeira em pelo menos uma valoração). Δ

Definição 3.13
Conjunto Satisfatível de Fórmulas

Se K for um conjunto de fórmulas e se existir um modelo $<W, R, S, \nu>$ tal que toda fórmula $X \in K$ seja satisfatível para esse modelo, então o conjunto K é dito satisfatível. Δ

Definição 3.14
Fórmula Válida.

Diz-se que uma fórmula X é **válida** se X for satisfeita em todos os modelos $<W, R, S, \nu>$. Δ

Notação: $\models X$.

Definição 3.15
Isomorfismo Entre NK-Estruturas

Sejam $\langle W, R, S \rangle$ e $\langle W', R', S' \rangle$ duas NK-estruturas quaisquer. Uma bijeção $f: W \rightarrow W'$ é

chamada de isomorfismo se, para quaisquer $w_1, w_2, w_3, w_4 \in W$, observar-se:

$w_1 \; R \; w_2$ se e somente se $f(w_1) \; R' \; f(w_2)$

$w_3 \; S \; w_4$ se e somente se $f(w_3) \; S' \; f(w_4)$ $\triangle$

Teorema 3.1

a) O terno $\langle \mathbb{Z}^2, R, S \rangle$ onde R e S satisfazem a : (i, j) R (i−1, j) $\forall i, j \in \mathbb{Z}^2$ e (i, j) S (i, j−1) $\forall i, \in \mathbb{Z}^2$ é uma NK-estrutura.

b) Toda NK-estrutura $\langle W, R, S \rangle$ é isomórfica a NK-estrutura $\langle \mathbb{Z}^2, R, S \rangle$.

Demonstração:

a) É possível estabelecer uma bijeção g: $\mathbb{Z}^2 \rightarrow \mathbb{Z}^2$, e como as relações R e S possuem as seguintes características (i, j) R (i−1, j) e (i, j) S (i, j−1) para todo i, j pertencente a $\mathbb{Z}^2$, correspondendo a definição de NK-estrutura. Portanto $\langle \mathbb{Z}^2, R, S \rangle$ é uma NK-estrutura.

b) Da definição de NK-estrutura, pois se $\langle W, R, S \rangle$ é uma NK-estrutura existe uma bijeção f:W→ $\mathbb{Z}^2$ e os pares de relações R, S, R e S obedecem as condições da definição 3.14 de isomorfismo entre NK-estruturas.▲

Lema 3.1

Seja $\langle W, R, S \rangle$ isomórfica através da bijeção f : W → W' a $\langle W', R', S' \rangle$. Seja ν uma valoração definida sobre $\langle W, R, S \rangle$ e ν' uma valoração definida sobre $\langle W', R', S' \rangle$ tais que para toda variável proposicional p, $\nu(p, w) = \nu'(p, f(w))$. Então, $\nu(X, w) = \nu'(X, f(w))$ para toda fórmula X.

Demonstração, por indução no grau da fórmula:

Passo 0: As variáveis proposicionais são fórmulas de grau zero e, por hipótese, satisfazem à tese do teorema.

Hipótese indutiva: Passo indutivo: Seja X uma fórmula de grau n e Y uma fórmula de grau m.

Prova-se agora que todas as fórmulas obtidas a partir destas fórmulas satisfaz à tese do teorema.

- A fórmula ¬X tem grau n+1. $\nu(\neg X, w) = t$ se e somente se $\nu(X, w) = f$. Mas $\nu(X,w)=\nu'(X,f(w))=f \Rightarrow \nu'(\neg X, f(w)) = t$. Concluindo-se que $\nu(\neg X, w) = t$ então $\nu'(\neg X, f(w))$. analogamente $\nu(\neg X, w) = f$ então $\nu'(\neg X, f(w))=f$.
- A fórmula $X \wedge Y$ tem grau n+m+1. $\nu((X \wedge Y), w) = t$, se e somente se $\nu(X, w) = t$ e $\nu(Y, w) =$

t. Mas $\nu(X, w) = \nu'(X, f(w)) = t$ e $\nu(Y, w) = \nu'(Y, f(w)) = t$ pela hipótese indutiva. Concluindo-se que se $\nu((X \wedge Y), w) = t$ então $\nu'((X \wedge Y), f(w)) = t$. Analogamente se $\nu((X \wedge Y), w) = f$ então $\nu'((X \wedge Y), f(w)) = f$.

- A fórmula $X \vee Y$ tem grau n+m+1. $\nu((X \vee Y), w) = t$, se e somente se $\nu(X, w) = t$ ou $\nu(Y, w) = t$. Então, tem-se que $\nu'(X, f(w)) = t$ ou $\nu'(Y, f(w)) = t$ se e somente se $\nu'((X \vee Y), f(w)) = t$. Daí $\nu((X \vee Y), w) = t$ se e somente se $\nu((X \vee Y), f(w)) = t$.
- A fórmula $X \rightarrow Y$ tem grau n+m+1. $\nu((X \rightarrow Y), w) = t$, se e somente se $\nu(X, w) f$ e $\nu(Y, w) =$ Então, pela hipótese indutiva $\nu'(X, f(w)) = f$ e $\nu'(Y, f(w)) = t$ se e somente se $\nu'((X \rightarrow Y), f(w)) = t$. Daí $\nu((X \rightarrow Y), w) = t$ se e somente se $\nu'((X \rightarrow Y), f(w)) = t$.
- A fórmula $X \leftrightarrow Y$ tem grau n+m+1. $\nu((X \leftrightarrow Y), w) = t$, se e somente se $\nu(X, w) = t$ e $\nu(Y, w) = t$ Pela hipótese indutiva tem-se que $\nu'(X, f(w)) = t$ e $\nu'(Y, f(w)) = t$ se e somente se $\nu'((X \leftrightarrow Y), f(w)) = t$. Daí $\nu((X \leftrightarrow Y), w) = t$ se e somente se $\nu'((X \leftrightarrow Y), f(w)) = t$.
- A fórmula DX tem grau n+1. $\nu(DX, w) = t$, se e somente se $\exists w' \in W$ tal que wSw' e $\nu(X, w') = t$, mas pela definição de isomorfismo e por ν' definida acima $\exists f(w') \in W'$ tal que $f(w) S' f(w')$ e, pela hipótese indutiva, $\nu'(X, f(w')) = t$. Daí $\nu(DX, w) = t$ se e somente se $\nu'(DX, f(w)) = t$.
- A fórmula GX tem grau n+1. $\nu(GX, w) = t$, se e somente se $\exists w' \in W$ tal que wRw' e $\nu(X, w') = t$, mas pela definição de isomorfismo e por ν' definida acima $\exists f(w') \in W'$ tal que $f(w) R' f(w')$ e, pela hipótese indutiva, $\nu'(X, f(w')) = t$. Daí $\nu(GX, w) = t$ se e somente se $\nu'(GX, f(w)) = t$. ▲

Teorema 3.2

Se uma fórmula é verdadeira em todo modelo $\langle \mathbb{Z}^2, R, S, \nu \rangle$ então ela é válida.

Demonstração:

Suponha, por absurdo, que uma fórmula X verdadeira em todo modelo $\langle \mathbb{Z}^2, R, S, \nu \rangle$ não seja válida. Portanto, existe uma valoração ν em algum modelo $\langle W, R, S, \nu \rangle$ onde $v(X) = f$. Mas pelo teorema 3.1 todas as NK-estruturas são isomórficas a $\langle \mathbb{Z}^2, R, S \rangle$, e pelo lema 3.1 tem-se que se duas NK-estruturas $\langle W, R, S \rangle$ e $\langle W', R', S' \rangle$ são isomórficas, então $\nu(X, w) = \nu'(X, f(w))$ para toda fórmula X. Logo não é possível que $v(X) = f$, provando por absurdo a tese. ▲

Na continuação deste trabalho, graças ao teorema 3.2, uma fórmula será considerada válida se ela for verdadeira para qualquer valoração da NK-estrutura $<\mathbb{Z}^2, R, S>$.

3.4 Tableau Analítico e Notação Unificada.

A partir deste ponto são apresentados conceitos da lógica modal NK sob o ponto de vista do tableau analítico (Smullyan, 1968).

O tableau é um procedimento de prova lógica (verifica-se se uma fórmula é, ou não, válida) que se apresenta na forma de árvores binárias (apêndice A) contendo sempre um número finito de ramos nos quais cada nó contém uma fórmula associada a um valor verdade. Este método conta com a facilidade de não utilizar o mecanismo da substituição como no método de prova por axiomas, mas se utiliza apenas de subfórmulas (para uma comparação entre os métodos de prova veja Costa, 1992).

Definição 3.16
Fórmula Assinalada

Sejam T e F dois símbolos formais. Por uma fórmula assinalada entende-se TX ou FX, onde X é uma fórmula. △

Descrição do Método Tableau Proposicional Clássico.

Inicia-se a construção da árvore de prova negando-se a fórmula que se deseja provar, ou seja, inserindo-se a fórmula a ser analisada no topo da árvore tendo como valor assinalado F (figura 3.1 abaixo), os outros nós da árvore serão acrescentados pela aplicação das regras tableau apropriadas (apêndice A para uma definição precisa de tableau), o objetivo é tentar falsear a fórmula que se está testando afirmando que esta é falsa e, a partir daí, aplicar-se todas as regras tableau apropriadas que forem possíveis. Caso todos os ramos abertos pela aplicação dessas regras contenham contradições, conclui-se que a fórmula testada é uma tautologia. Isso acontece porque é impossível falsear a fórmula testada pois partiu-se da hipótese de que ela era falsa chegando-se sempre a uma contradição. Se algum ramo do tableau não contiver uma contradição após a aplicação de todas as regras tableau possíveis isso significa que existe uma valoração (ou modelo) que falseia a fórmula testada o que permite concluir que ela não é uma tautologia.

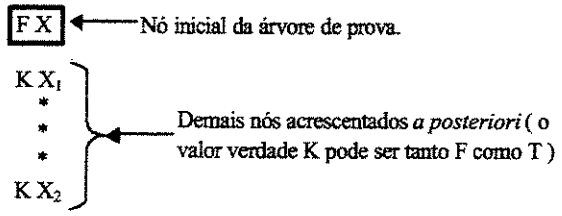


figura 3.1
Ramo de um tableau para a fórmula X.

Tipos de Regras.

As regras para o **tableau proposicional clássico** são divididas em dois tipos: As que causam bifurcações na árvore de prova e as que não causam (posteriormente serão introduzidas as regras relativas às fórmulas modais). As regras que não causam bifurcações na árvore são conhecidas como regras do tipo α, enquanto as regras que causam bifurcações na árvore de prova são conhecidas como regras do tipo β. As regras do tipo α causam a inserção no ramo de todos os componentes ou subfórmulas ligadas pelo conectivo mais externo da fórmula sobre a qual se aplicou a regra α apropriada. No caso da aplicação da regra β cria-se uma bifurcação no ramo onde cada ramo dessa bifurcação contém um dos componentes da fórmula sobre a qual se aplicou a regra β apropriada Graficamente, essas regras podem ser vistas conforme mostram as figuras 3.2 e 3.3 abaixo:

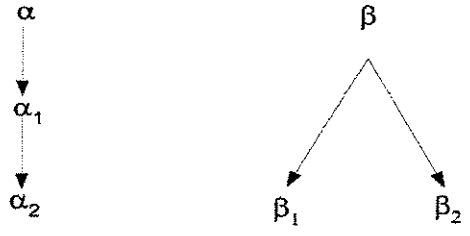


figura 3.2 **figura 3.3**
Esquema para as regras α e β.

α-fórmula.

Vista sob a ótica dos valores-verdade de seus componentes, as fórmulas do tipo α são apresentadas na tabela 3.1 a seguir. Cada α-fórmula dá origem a uma regra do tipo α.

Regras	Fórmula	Componentes	
α1	$T(X \wedge Y)$	T X	T Y
α2	$F(X \vee Y)$	F X	F Y
α3	$F(X \rightarrow Y)$	T X	F Y
α4	$F\neg X$	T X	

Tabela com as regras do tipo α

tabela 3.1

É imediato concluir, a partir da semântica das fórmulas da tabela 3.1 que para que uma α-fórmula posa ser afirmada seus componentes devem ser afirmados conjuntamente.

β- fórmula.

Idêntico às fórmulas α vistas anteriormente, sob a ótica dos valores-verdade de seus componentes, as fórmulas do tipo β são apresentadas na tabela 3.2 abaixo. Como anteriormente, cada β-fórmula dá origem a uma regra do tipo β.

Regras	Fórmula	Componentes	
β1	$F(X \wedge Y)$	F X	F Y
β2	$T(X \vee Y)$	T X	T Y
β3	$T(X \rightarrow Y)$	F X	T Y
β4	$T \neg X$	F X	

Tabela com as regras do tipo β

tabela 3.2

Analogamente às α-fórmulas, a tabela 3.2 permite concluir que uma β-fórmula é afirmada somente se seus componentes forem afirmados disjuntamente.

Um exemplo de prova tableau para um teorema, no caso, $\neg(A \vee B) \rightarrow (\neg A \wedge \neg B)$ tem a sua árvore de prova apresentada na figura 3.4 a seguir:

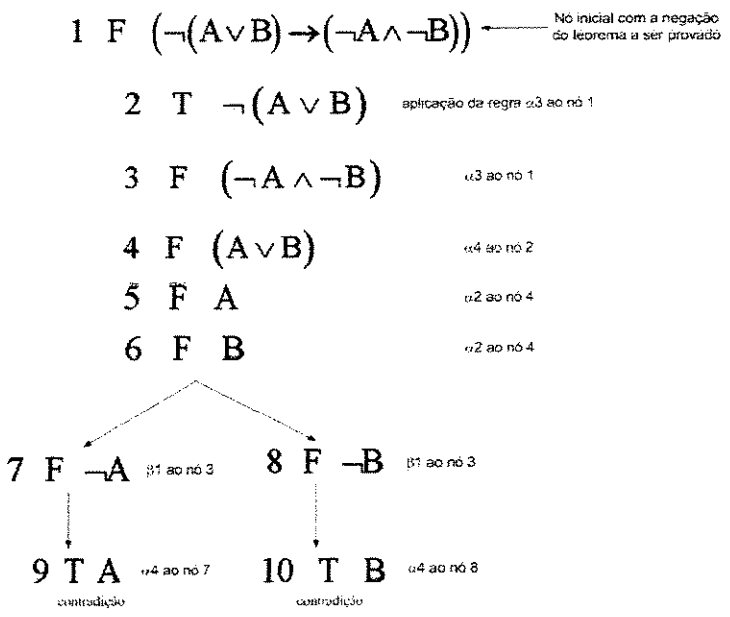


figura 3.4

Prova tableau para a fórmula ¬(A∨B)→(¬A∧¬B)

Conforme se pode analisar do exemplo anterior, a fórmula testada é, realmente, uma tautologia, pois os dois únicos ramos {1, 2, 4, 5, 6, 7, 9} e {1, 2, 3, 4, 5, 6, 8, 10} do tableau estão **fechados**, ou seja, possuem duas fórmulas iguais com valores de verdade diferentes, no caso T A e F A no ramo da esquerda e T B e F B no ramo da direita. Isso mostra que não há maneira de falsear a fórmula testada ou, em outras palavras não há uma valoração para os seus componentes (no caso A e B) para a qual construída uma tabela verdade dê como valor verdade de (¬(A ∨ B) → (¬A ∧ ¬B)) o valor f. Se porventura algum dos ramos não apresentasse uma contradição, teríamos então uma valoração (ou um **modelo**) que falsearia a fórmula demonstrando que ela não é uma tautologia. Pode-se visualizar essa situação modificando-se a tautologia do exemplo anterior para (¬(A ∨ B) → (¬A ∧ B)) e executando-se o mesmo procedimento anterior que gera a árvore seguinte:

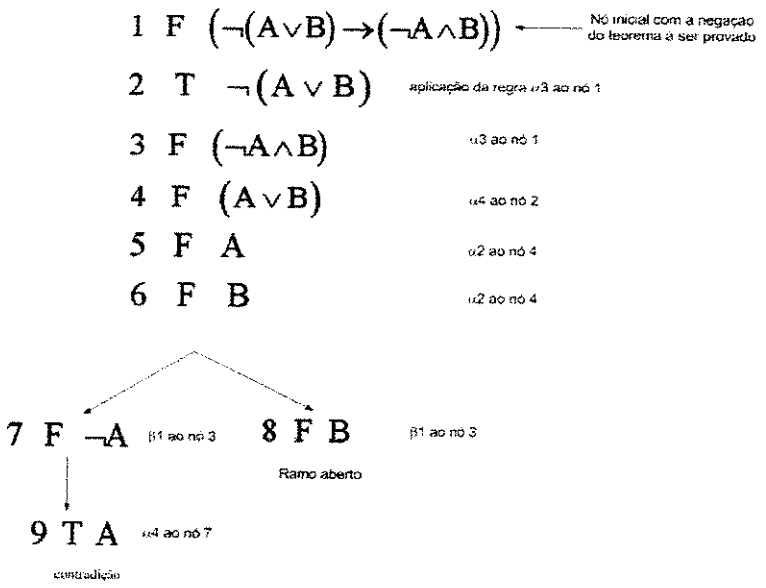


figura 3.5

Tableau para a fórmula ¬(A∨B)→(¬A∧B) retornando, pelo ramo da direita, uma valoração que falseia a fórmula testada.

Conforme se pode observar acima o ramo da direita não apresenta a contradição do exemplo anterior, o que se configura num modelo, ou valoração que falseia a fórmula testada. Observando-se o ramo aberto, a valoração que permite falsear a fórmula testada consiste em atribuir o valor verdade f A e a B, cujo resultado é apresentado na tabela 3.3.

	$(A \vee B)$	$\neg(A \vee B)$	$(\neg A \wedge B)$	$\neg(A \vee B) \rightarrow (\neg A \wedge B)$
$v(A) = f$ $v(B) = f$	f	t	f	f

Tabela-verdade com a valoração retornada pelo ramo aberto da figura 3.7

tabela 3.3

Pela tabela acima vê-se que o tableau "gerou" um modelo que falseia a fórmula testada, confirmando a informação obtida via tableau. O ramo acima é dito um ramo satisfatível, pois o seu conjunto de fórmulas também o é, e como, pelo menos, um ramo do tableu é satisfatível, então tableau também é dito satisfatível (veja as definições abaixo).

Definição 3.17
Ramo Satisfatível de um Tableau

Um ramo de um tableau é satisfatível se e somente se o conjunto de fórmulas contido no ramo é satisfatível.△

Definição 3.18
Tableau satisfatível

Um tableau é satisfatível se pelo menos um de seus ramos for satisfatível. △

Tableau para a Lógica-NK

Definição 3.19
Prefixo.

Um prefixo é um par (i, j) tal que $i, j \in \mathbb{Z}$. △

Definição 3.20
Fórmula Prefixada

Uma fórmula prefixada (i, j) X constitui-se de um prefixo (i, j) ao lado de uma fórmula X. De maneira análoga uma fórmula prefixada assinalada (i, j) Z constitui-se num prefixo (i, j) acompanhado da fórmula assinalada Z. △

Definição 3.21
Prefixo Usado.

Um prefixo $\sigma = (i, j)$ pode ser dito usado num ramo do tableau NK (para fórmulas prefixadas) se e somente se σ já acorreu anteriormente no ramo. △

Observação:

Conforme foi dito acima um prefixo representa um nome para um mundo possível, então têm-se que uma fórmula σX é verdadeira numa dada interpretação, se X for verdadeira no mundo cujo nome é σ.

Definição 3.22
R-acessibilidade e S - acessibilidade.

Um prefixo $\sigma' = (r, s)$ é dito R - acessível a partir de outro prefixo $\sigma = (i, j)$ se e somente se $r = i-1$ e $s = j$. De maneira análoga $\sigma' = (r, s)$ é dito S - acessível a partir de outro prefixo $\sigma = (i, j)$ se somente se $r = i$ e $s=j-1$. △

Como cada prefixo pertence a $\mathbb{Z}^2$, têm-se que a partir de qualquer $\sigma \in \mathbb{Z}^2$ sempre existirá um prefixo σ', S-acessível ou um prefixo σ', R-acessível.

ν-fórmula e π-fórmula.

O tableau apresentado até o momento se refere ao cálculo proposicional, mas o objetivo é um **tableau modal NK**, o que torna necessário introduzir mais **duas regras** referentes aos operadores modais G e D previamente definidos. As fórmulas do tipo GX dão origem às regra ν. Esta regra estabelece que se uma fórmula deste tipo é verdadeira ou falsa num mundo (i, j) então a fórmula X deverá ser igualmente verdadeira ou falsa no mundo R-acessível a (i, j). A regra π é introduzida de modo análogo para o operador D em combinação com a relação S. As tabelas 3.4 e 3.5 sumarizam essas regras.

ν-fórmula	componente
ν	ν_0
T Gx	T x
F Gx	F x

π-fórmula	componente
π	π_0
T Dx	T x
F Dx	F x

tabela 3.4 **tabela 3.5**

Tabelas de valores para as regras ν e π.

Sob a ótica da inserção dos nós no tableau as regras podem ser vistas abaixo, com $k > 0$ e $n > 0$ indicando linhas do tableau e os índices entre parênteses indicando a manipulação dos nomes dos mundos possíveis pelas regras ν e π.

k	ν	(i,j)	k	π	(i,j)
k+n	ν_0	(i-1,j)	k+n	π_0	(i,j-1)

figura 3.6 **figura 3.7**

Aplicação das regras ν e π no tableau.

Conseqüência Lógica e Dedutibilidade.

No cálculo proposicional clássico diz-se que uma fórmula X é **conseqüência lógica** de um conjunto Γ de fórmulas sempre que qualquer valoração ν que tornar as fórmulas do conjunto Γ verdadeiras também o fizer com X.

Notação: $\Gamma \models X$ (X é conseqüência lógica de Γ).

Em termos de tableau, se $S = \{X_1, X_2, X_3 \ldots X_n\}$ é um conjunto de fórmulas não assinaladas e X é, também uma fórmula não assinalada tem-se: $S \vdash_{dt} X$ (lê-se "X é dedutível tableau via S") se e somente se existir um tableau **fechado** e **completado** (apêndice A) iniciado por FX onde são inseridas as fórmulas de S como verdadeiras em qualquer prefixo .

3.5 Correspondência Entre a Lógica Modal NK e o Dióide *M*

Neste item será apresentada a correlação entre a lógica NK e o dióide *M*, para tanto será definida a função H, que será utilizada para definir uma I-interpretação. Teoremas importantes para a modelagem de grafos a eventos serão apresentados .

Definição 3.23
Função de Ligação.

Seja uma função que associa cada elemento do conjunto de variáveis proposicionais P da lógica NK a um único elemento do dióide *M*. Essa função será chamada de função de ligação e denotada por H. Seja, então a função H:P $\rightarrow$ *M* e M_H a imagem dessa função. Como H é uma função injetora em *M*, então pode-se definir uma função I:$M_H \rightarrow$ P , tal que I(H(p))=p para qualquer variável p (I é a inversa de H quando se restringe o conjunto *M* ao conjunto M_H conforme mostra a figura 3.8 a seguir) . $\triangle$

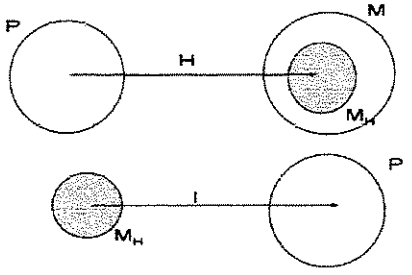


figura 3.8
Funções H e i.

Definição 3.24.
I-interpretação.

Dada uma função de ligação H e a correspondente função I, chama-se uma I-interpretação para a lógica NK a uma valoração ν na qual, para todo a $\in M_H$ e para todo i,j $\in \mathbb{Z}$:

$\nu((i, j), I(a)) = t$ se e somente se $a \geq \gamma^i \delta^j$. $\triangle$

Uma I-interpretação atribui valor verdade para todas as variáveis proposicionais em todos os elementos de W determinando, univocamente para todas as fórmulas da lógica, seu valor-verdade (figura 3.9 abaixo).

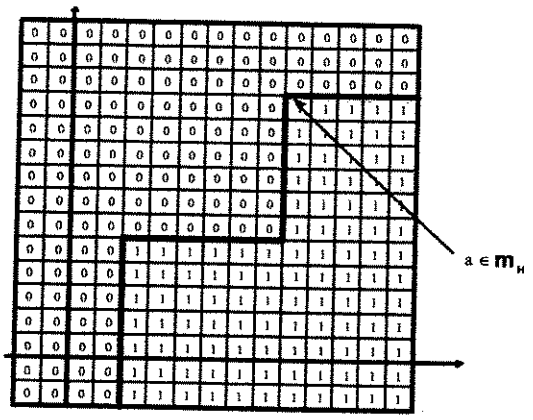


figura 3.9
Um elemento $a \in M_H$.

Definição 3.25.
Fórmula i-válida.

Uma fórmula X de NK é i-válida ($\models_i$ X) se e somente se ela for verdadeira em todas as I-interpretações. △

A partir de agora serão apresentados alguns resultados da lógica NK que permitem associar uma especificação de desempenho expressa como uma igualdade (ou desigualdade) linear em M. Caso a especificação seja expressa por uma igualdade, será obtida uma fórmula bicondicional da lógica NK que corresponde a ela. Se a especificação for expressa por uma desigualdade obter-se-á uma fórmula condicional correspondente na lógica NK. Finalmente será demonstrado que uma especificação será considerada satisfeita se a fórmula NK correspondente a ela for satisfeita para toda interpretação da lógica NK que satisfaça o conjunto de fórmulas obtido a partir do GET no qual se está verificando a especificação. Estes resultados foram originalmente demonstrados por Magossi (1998)

Resultados da Lógica NK (Magossi, 1998).

Dada uma I-interpretação:

1. Para a, b e $(a \oplus b) \in M_H$ então, a seguinte fórmula de NK é válida:
   $I(a \oplus b) \leftrightarrow (I(a) \vee I(b))$.
2. Para a, $\gamma^n \delta^d a \in M_H$ e n, d $\in \mathbb{Z}^+$ então, é válida a seguinte fórmula:

$G^n D^d I(a) \leftrightarrow I(\gamma^n \delta^d a)$

3. Para qualquer $a \in M_H$, as duas fórmulas seguintes são verdadeiras:

   $GI(a) \rightarrow I(a)$

   $I(a) \rightarrow DI(a)$

4. Para qualquer $a \in M_H$, $a \geq b$ se e somente se a seguinte fórmula da lógica NK for válida:

   $I(b) \rightarrow I(a)$

5. Se $\mathcal{A}$ é uma valoração da lógica NK tal que as fórmulas $Gp \rightarrow p$ e $p \rightarrow Dp$ sejam satisfeitas para todas as variáveis proposicionais em todos os mundos sob a valoração $\mathcal{A}$, então a valoração $\mathcal{A}$ é uma I-interpretação.

6. Sejam x_i, com $i = 1,2,...,n$ elementos do dióide M, então o conjunto Θ de equações do tipo $\Theta = \left\{ x_i = \sum_{j=1}^{n} \gamma^{m_{ij}} \delta^{d_{ij}} x_j \, , i=1,2,3,..,n \right\}$ é satisfeito se e somente se o conjunto Ω de NK fórmulas notado por:

$$\Omega = \left\{ \begin{array}{c} p_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} p_j \\ Gp_i \rightarrow p_i \\ p_i \rightarrow Dp_i \\ (i=1,2,3,...n) \end{array} \right\} \text{for verdadeiro em todos os mundos de alguma valoração.}$$

Temos então que **o conjunto Ω das fórmulas da lógica NK associado a um GET** é descrito do seguinte modo:

$$p_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} p_j \; (i = 1, 2, \ldots n)$$
$$Gp_i \rightarrow p_i \; (i=1,2,...n)$$
$$p_i \rightarrow Dp_i \; (i=1,2,...n)$$

Com m_{ij} e d_{ij} a marcação e o atraso iniciais do lugar com transição de entrada t_j e de saída t respectivamente e $p_i = I(t_i)$ e $p_j = I(t_j)$, com $i = 1,2,..,n$ e $j = 1,2,...,n$. O exemplo da rede na figura 3.10 abaixo ilustra essa representação.

$$I(t_0) \leftrightarrow \bigvee_{j=1}^{k} G^{m_j} D^{d_j} I(t_j)$$

$$GI(t_j) \rightarrow I(t_j)$$

$$I(t_j) \rightarrow DI(t_j) \ (j=1,2,...k)$$

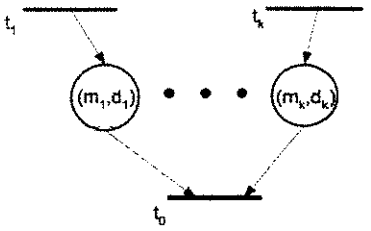


figura 3.10
Esquema de um GET

Como foi visto no capítulo 2, é possível estabelecer um modelo algébrico com equações no dióide M para qualquer GET, sendo os elementos de M subconjuntos do plano $\mathbb{Z}^2$. Neste capítulo viu-se que os elementos de $\mathbb{Z}^2$ podem ser associados ao conjunto de mundos possíveis através da semântica da lógica NK, e que uma transição do GET, associada a um elemento do dióide M pode, se as condições de uma I-interpretação forem satisfeitas, ser associado a uma variável proposicional da lógica NK. Cria-se, então, uma correspondência entre os elementos do dióide M e os elementos da lógica NK conforme a tabela abaixo:

Elementos da lógica	GET/Dióide M
$I(t_k)$ (variável proposicional)	t_k (transição)
$V((i,j), I(t_k)) = t$	$t_k \geq \gamma^i \delta^j$

Tabela 3.6
Correspondência entre os elementos da lógica NK e do dióide M

No próximo capítulo serão apresentados resultados que permitirão construir um procedimento algorítmico do tipo tableau para verificar se a fórmula que expressa uma especificação é conseqüência lógica do conjunto de fórmulas correspondente ao GET (ítem 6 dos resultados acima).

A construção de um tableau utilizando as informações da rede conforme mostrado acima e através do procedimento algorítmico permite verificar especificações para essa mesma rede, se a especificação é válida o tableau se apresentará fechado em todos os ramos, já se a especificação não for válida, ao negá-la no primeiro nó do tableau este retornará exatamente a valoração (ou modelo) que falseia a especificação. Veja o exemplo de uma verificação de especificação para a rede da figura 3.11 abaixo.

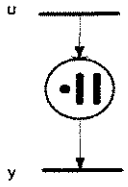


figura 3.11
GET com saída $\gamma\delta^2$

Fórmulas: $y \leftrightarrow GD^2u$

$y \rightarrow Dy$

$Gy \rightarrow y$

$Gu \rightarrow u$

$u \rightarrow Du$

A especificação a ser avaliada é $y \rightarrow Du$. Graficamente pode-se ver que esta especificação não é satisfeita (figura 3.12), pois o cone a sudeste que representa a saída y não é menor ou igual ao cone a sudeste que representa δu.

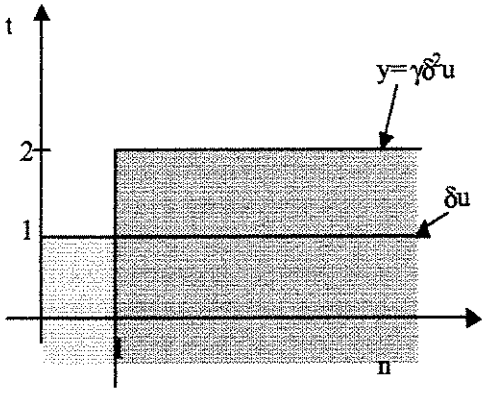


figura 3.12
Saída do grafo da figura 3.11.

Portanto, o tableau deve retornar um ramo aberto para esta especificação, o que é confirmado pela figura 3.16 a seguir. Nestas figuras, por simplicidade, descreve-se o tableau como uma matriz de células, onde cada célula representa um mundo possível e onde as fórmulas prefixadas por algum mundo são inseridas na célula correspondente ao mesmo mundo.

Além disso, inicia-se a construção do tableau de um ponto genérico (i, j), significando que não se sabe a priori em que ponto a especificação não é satisfeita. Conforme será visto no próximo capítulo, a determinação de onde se situa o ponto (0, 0) (e, por conseguinte, qual é o ponto (i, j)) é um aspecto

importante da utilização prática do algoritmo tableau.

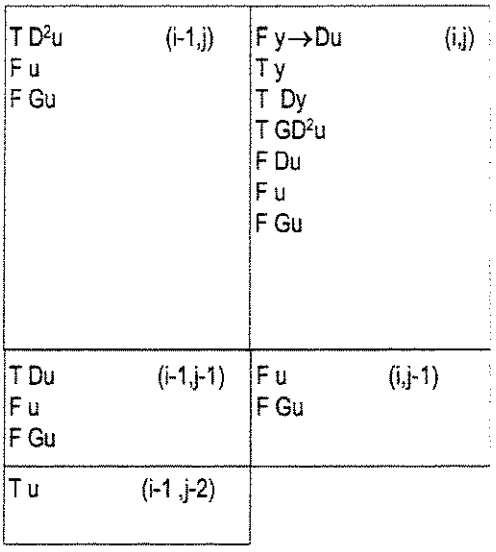


figura 3.13

Ramo tableau para o GET da figura 3.11

Observa-se claramente na figura 3.13 acima que o tableau retornou um modelo que falseia a especificação checada, pois retornou um ramo aberto. No caso de uma especificação diferente como $y \rightarrow D^2u$ têm-se o seguinte ramo do tableau.

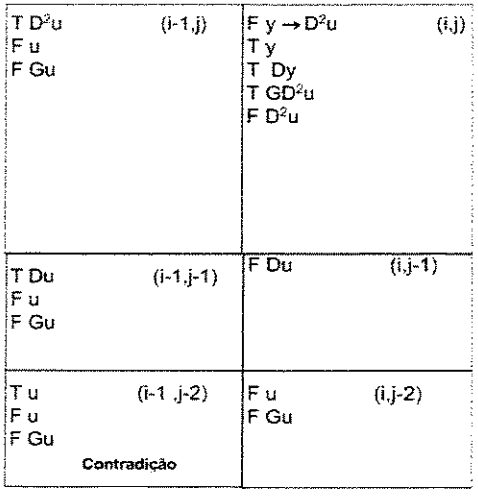


figura 3.14

Tableau fechado para a especificação $y \rightarrow D^2u$ para o GET da figura 3.14

Conforme se pode ver na figura 3.14, o tableau não retornou um modelo, mas sim um ramo fechado (contradição no mundo (i–1,j–2)) o que mostra a impossibilidade de falsear a especificação pois, de

fato, y é menor que $\delta^2 u$. Graficamente essa situação pode ser vista na figura 3.15 abaixo:

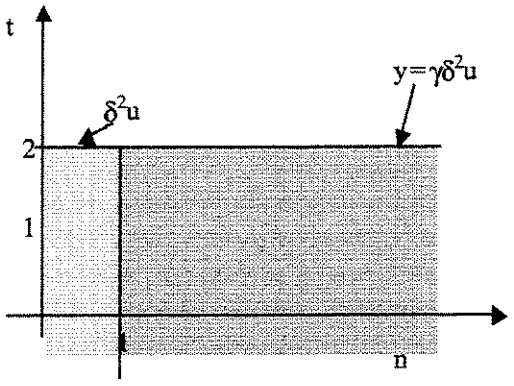


figura 3.15

Especificação do ramo tableau da figura 3.11 vista no plano n×t confirmando o ramo fechado.

Na figura 3.19 abaixo têm-se outro grafo que pode ser descrito, conforme já foi visto, pelas fórmulas da lógica nk da seguinte maneira:

$z \leftrightarrow G^2D^2z \vee u$

$z \rightarrow Dz$

$Gz \rightarrow z$

$y \leftrightarrow G^2D^2y \vee GDu$

$y \rightarrow Dy$

$Gy \rightarrow y$

$u \rightarrow Du$

$Gu \rightarrow u$

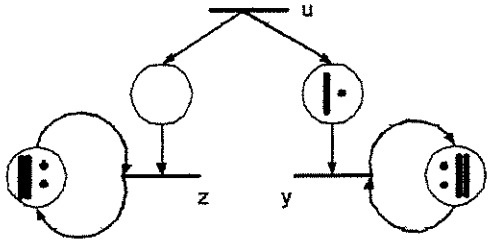


figura 3.16

GET com duas saídas z e y.

A trajetória de ambas as saídas y e z pode ser vista na figura 3.17 abaixo:

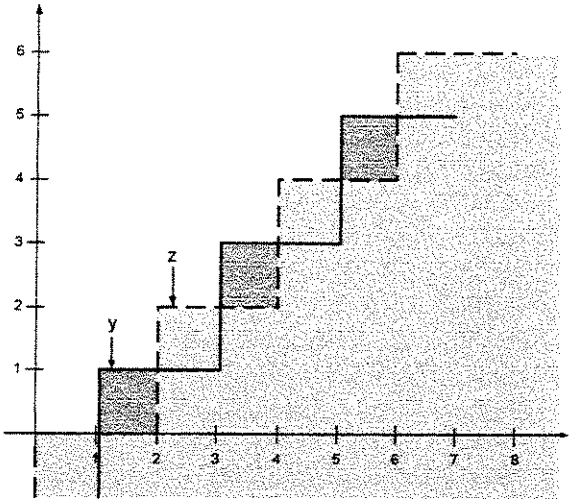


figura 3.17

Trajetória no plano para o GET da figura 3.16

A especificação $z \rightarrow y$ pode ser verificada pelo tableau, e um ramo aberto pode ser visto abaixo:

(i-2, j) F y F Gy F GDu v G^2D^2y F GDu F G^2D^2y F GD^2y T D^2z	(i-1,j) F y F Gy F GDu v G^2D^2y F GDu F G^2D^2y T GD^2z	(i, j) F $z \rightarrow y$ F y F Gy F GDu v G^2D^2y F GDu F G^2D^2y T z T u v G^2D^2z T G^2D^2z
(i-2,j-1) F u F Gu F G^2Dy F GDy T Dz	(i-1,j-1) F Gu F G^2Dy T GDz	(i,j-1) F Gu F G^2Dy T z T G^2Dz
(i-2,j-2) T z T u v G^2D^2z T u	(i-1,j-2) F G^2y T Gz	

figura 3.18

Ramo tableau para a especificação $z \rightarrow y$ do GET da figura 3.16

Conforme se pode ver pelo ramo acima, a especificação $z \rightarrow y$ pode ser, de fato, falseada, fato confirmado pelo exame das trajetórias na figura 3.17.

3.6 Conclusão

Neste capítulo discutiu-se a lógica modal NK, aspectos de sua sintaxe e também da semântica dos mundos possíveis. Viu-se também que a lógica NK é equivalente ao dióide M apresentado no capítulo 2 e como a verificação de especificações para um GET pode ser feita através da confirmação de que a fórmula que expressa a especificação a ser verificada é conseqüência lógica do conjunto de fórmulas NK para o GET. Discutiu-se o tableau proposicional clássico e como ele pode ser utilizado para verificar a existência de um modelo ou interpretação para uma dada fórmula extendendo-se a seguir este mesmo tableau através das regras apropriadas tornando-o capaz de verificar a conseqüência lógica na lógica NK. Através dos resultados que estabelecem a correspondência entre lógica NK e o dióide M essa verificação de conseqüência lógica corresponde a verificar especificações para um GET utilizando-se a lógica modal NK. No capítulo seguinte serão apresentados alguns resultados visando dotar o mecanismo tableau de um critério de parada eficiente, assim como de um critério de identificação dos ramos abertos associados à solução mínima, única e causal procurada.

CAPÍTULO 4

Conectividade, Terminalidade e Causalidade.

4.1 Introdução.

Este capítulo apresenta conceitos como terminação, conexão e causalidade, mostrando teoremas que os ligam e que auxiliam a responder algumas questões sobre o o tableau apresentado no capítulo anterior, como: Quando parar a verificação? Seria possível o tableau retornar uma solução **causal e não mínima**? Caso o tableau NK retorne uma solução assim como identificá-la? O algoritmo apresentado por Magossi e Santos Mendes, (1998) aplicável a um ramo aberto é uma tentativa de responder a esta última questão. Um exemplo de solução não mínima e não causal (definição 4.6) é dado pelo ramo da figura 4.2 sendo a especificação y → Du verificada para as fórmulas do GET da figura 4.1 cuja saída é dada por y = $\gamma y \oplus u$. Na figura 4.3 tem-se uma representação no plano n×t da solução representada pelo ramo da figura 4.2 que corresponde a $\gamma^{-\infty}\delta$. Torna-se necessário portanto, um critério de identificação dos ramos do tableau associados a características indesejadas como a não-causalidade e não-minimalidade assim como a características desejáveis na solução representada pelo ramo do tableau como minimalidade, causalidade, unicidade e impulsividade. Esse critério será dado pelos teoremas 4.4, 4.5 e 4.6, tornando possível ao tableau identificar os ramos associados à solução desejada, ou seja, mínima, causal, única e impulsiva e, ao mesmo tempo, descartar os ramos associados a soluções não desejadas como as não-causais e não mínimas. O conceito de terminação (definição 4.3) de um ramo se mostrará diretamente relacionado ao conceito de solução mínima, assim como ao conceito de solução causal tornando-se o principal critério de seleção de ramos pelo algoritmo tableau. A definição de conectividade (definição 4.2) apresentar-se-á como central para o desenvolvimento dos teoremas 4.1 e 4.2 assim como o lema 4.2. Os resultados apresentados garantem para o algoritmo proposto ao fim do capítulo propriedades que o tornam capaz de retornar uma solução que seja única, causal, mínima e impulsiva.

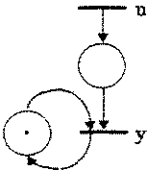


figura 4.1
GET com saída y = $\gamma y \oplus u$

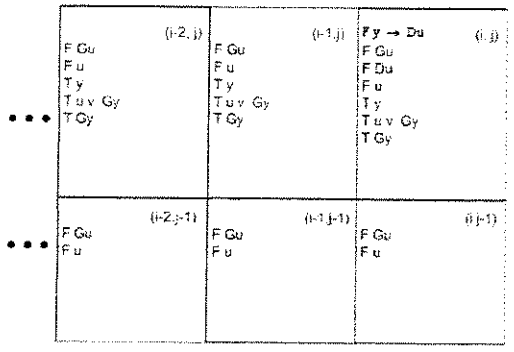


figura 4.2

Ramo tableau para o GET da figura 4.1 com especificação y→Du

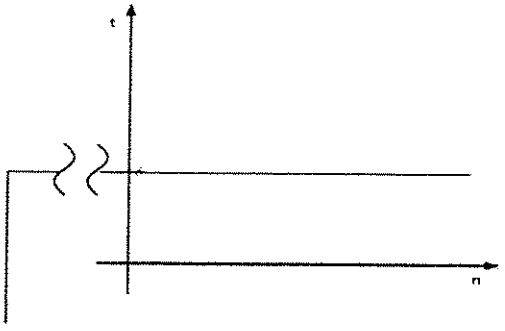


figura 4.3

Representação da solução não-mínima e não-causal dada pelo ramo da figura 4.2

4.2 Conceitos Relativos á Finitude dos Ramos de um Tableau

Definição 4.1
Vértice.

Dado um ramo aberto em um tableau NK, um mundo (i, j) é chamado de vértice de uma variável proposicional x se ocorrer a seguinte situação (figura 4.4): △

$$\begin{cases} Tx\,(i, j) \\ Fx(i-1, j) \\ Fx(i, j+1) \end{cases}$$

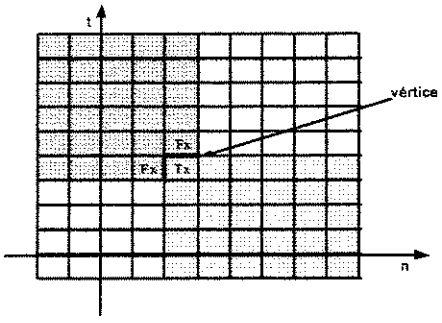


figura 4.4
Vértice de uma variável x

Definição 4.2
Conexão.

Dado um ramo aberto de um tableau NK, um vértice (i, j) de uma variável x_p está conectado a um vértice (k,l) de outra variável x_1 se as seguintes condições forem satisfeitas:

a) Existe um subconjunto de Ω com as fórmulas:

$$\begin{aligned} x_p &\leftrightarrow G^{n_{p-1}} D^{m_{p-1}} x_{p-1} \bigvee y_{p-1} \\ x_{p-1} &\leftrightarrow G^{n_{p-2}} D^{m_{p-2}} x_{p-2} \bigvee y_{p-2} \\ &\vdots \\ x_2 &\leftrightarrow G^{n_1} D^{m_1} x_1 \bigvee y_1 \end{aligned}$$

onde : $y_i = \bigvee_j G^{\alpha_{ij}} D^{\beta_{ij}} x_{ij}$, x_{ij} são variáveis proposicionais e α_{ij} e β_{ij} são os expoentes da seqüência $y_{p-1}, y_{p-2}, \cdots y_1$.

b) Existem os seguintes mundos possíveis:

$$(i, j), (i-n_{p-1}, j-m_{p-1}), \ldots, (i-\sum_{r=1}^{p-1} n_{p-r}, j-\sum_{r=1}^{p-1} m_{p-r}) = (k, l)$$

onde:

$$Tx_p \quad (i, j)$$
$$Tx_{p-1}\ (i - n_{p-1}, j - m_{p-1})$$
$$\vdots$$
$$Tx_1\ (i - \sum_{r=1}^{p-1} n_{p-r},\ j - \sum_{r=1}^{p-1} m_{p-r}) = (k, l)\ \Delta$$

Como as variáveis proposicionais nomeiam transições da rede é possível visualizar a situação descrita acima num grafo a eventos conforme a figura 4.5:

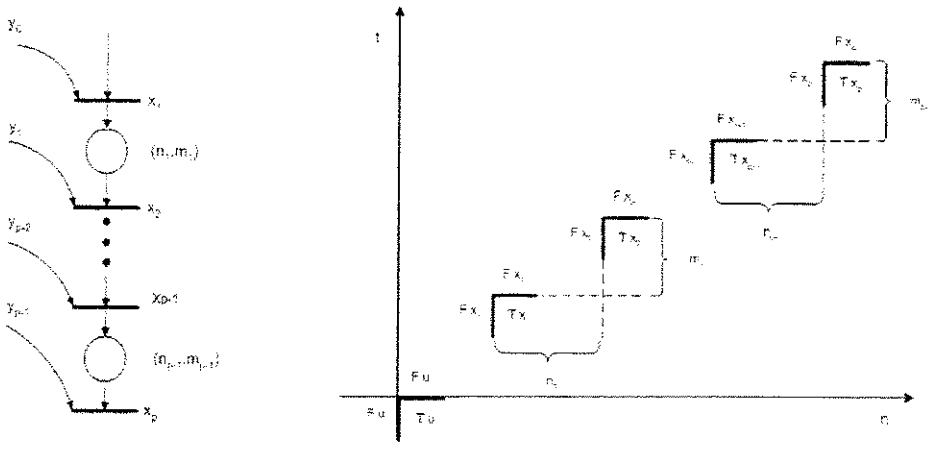


figura 4.5

Conexão entre variáveis e a situação correlata no GET

Observações :

1- O par ordenado $\left(\sum_{r=1}^{p-1} n_{p-r},\ \sum_{r=1}^{p-1} m_{p-r}\right) = (i-k,\ j-l)$ é chamado distância de conexão com $i \geq k$ e $j \geq l$.

2- A relação definida acima é transitiva mas não simétrica.

3- Se um vértice de uma variável está conectado a outro vértice da mesma variável então esse vértice é dito auto-conectado.

Definição 4.3
Terminação

Num ramo aberto do tableau NK uma variável proposicional X é dita terminada num mundo (i,j) se ocorrer a seguinte situação (figura 4.6):

1 (i, j) é um vértice

2 $F\,X\,(i-1, j-\beta), \forall\beta > 0.$ $\triangle$

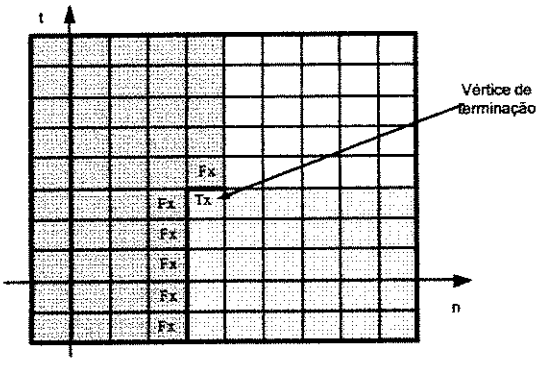


figura 4.6
Situação de terminação de uma variável

Definição 4.4
Variável Impulsivamente Terminada.

Num ramo aberto do tableau NK, uma variável proposicional X, terminada em (i, j) é dita impulsivamente terminada se a seguinte situação ocorrer (figura 4.7):

$FX\,(i+\alpha, j+1)\ \forall\alpha > 0.$ $\triangle$

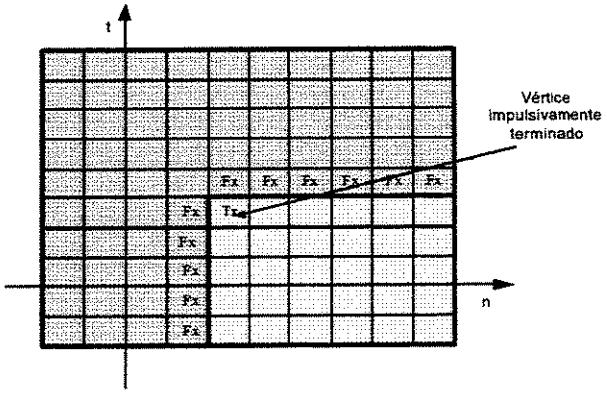


figura 4.7
Variável impulsivamente terminada

Definição 4.5
Ramo Terminado

Um ramo aberto do tableau NK será dito **terminado** se nele ocorrer a variável de entrada

impulsivamente terminada. △

Definição 4.6
Causalidade (Bacelli,. et al, 1992)

Um elemento h de $m_{in}^{ax}\|\gamma,\delta\|$ é causal se $h = \epsilon$ ou se val(h) ≥ 0 e $h \geq \gamma^{val(h)}$. Ou, conforme se vê em Cottenceau (1999), h é causal se h se encontra no **quadrante nordeste** do plano n×t (acima do eixo-n e a direita do eixo-t). △

Nota: Esta definição será utilizada na prova do teorema 4, que garante a causalidade das soluções encontradas via tableau.

4.3 Relações Entre os Conceitos de Conexão, Terminação e Causalidade.

Teorema 4.1

Todo vértice de toda variável, exceto a variável de entrada está conectado a algum vértice de alguma variável.

Demonstração:

Seja uma variável qualquer (diferente da entrada) x_l, para a qual deve existir, em Ω, uma fórmula da forma

1. $x_l \leftrightarrow \bigvee_{k=1}^{p} G^{\alpha_k} D^{\beta_k} x_k$.

Seja (i, j) um vértice da variável x_l, logo:

$$Tx_l(i, j)$$
$$Fx_l(i-1, j)$$
$$Fx_l(i, j+1)$$

Seja x_m uma das variáveis do lado direito de 1: nos mundos (i–1, j) e (i, j+1) tem-se que: $Fx_l \rightarrow FG^{\alpha_m} D^{\beta_m} x_m$. Logo, nos mundos $(i-\alpha_m-1, j-\beta_m)$ e $(i-\alpha_m, j-\beta_m+1)$ tem-se $F\,x_m$. No mundo (i, j) tem-se: Tx_l, portanto, para algum $m \in \{1, \ldots p\}$ tem-se $T\,G^{\alpha_m} D^{\beta_m} x_m$ no mundo (i, j) o que implica que temos Tx_m no mundo $(i-\alpha_m, j-\beta_m)$. Concluindo, $\exists x_m$, tal que

$$T\,x_m\,(i-\alpha_m, j-\beta_m)$$
$$F\,x_m\,(i-\alpha_m-1, j-\beta_m)$$
$$F\,x_m\,(i-\alpha_m, j-\beta_m+1)$$

logo, (i–α_m, j–β_m) é um vértice da variável x_m. As condições de conexão entre os vértices (i, j) (i–α_m,j–β_m) das variáveis x_l e x_m são verificadas, pois (figura 4.8):

a) $x_l \leftrightarrow G^{\alpha_m} D^{\beta_m} x_m \bigvee y_l$ onde $y_l = \bigvee_{k=1,k\neq m}^{p} G^{\alpha_k} D^{\beta_k} x_k$

b) $\exists$ T x_l (i, j) que é vértice de x_l.

$\exists$ T x_m (i–α_m, j–β_m) que é vértice de x_m. ▲

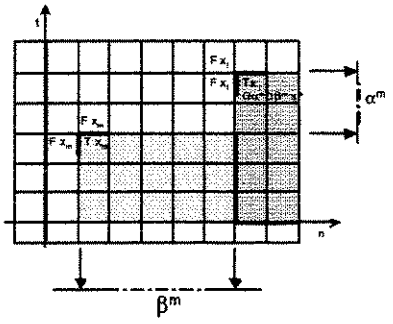


figura 4.8

Conexão entre duas variáveis

Lema 4.2

Num ramo terminado de um tableau NK, cada vértice de terminação de uma variável é conectado ao único vértice de terminação da entrada.

Demonstração:

Seja a variável x_l, diferente da entrada, com vértice no mundo (i, j). Suponha, por absurdo, que esse vértice esteja conectado a a mais de um vértice de terminação de entrada. Logo, existem, pelo menos duas fórmulas fl e f2 de Ω onde fl = $x_l \leftrightarrow G^{\alpha_m} D^{\beta_m} x_m \bigvee y_l$ e f2 =$x_l \leftrightarrow G^{\alpha_{m_1}} D^{\beta_{m_1}} x_{m_1} \bigvee$ y no caso de essa conexão aos vértices de terminação ser via transitividade (passando por x_m e x_{m_1}). Se essa conexão aos vértices de entrada for direta, tem-se fl = $x_l \leftrightarrow G^{\alpha_m} D^{\beta_m}$ u e f2=$x_l \leftrightarrow G^{\alpha_{m_1}} D^{\beta_{m_1}}$ u. Em ambos os casos tem-se a possibilidade da aplicação de um resultado do cálculo proposicional clássico:

$$x_l \leftrightarrow G^{\alpha_m} D^{\beta_m} u$$
$$x_l \leftrightarrow G^{\alpha_{m_1}} D^{\beta_{m_1}} u$$
$$\vdots$$
$$(2)\text{-}\quad x_l \leftrightarrow G^{\alpha_m} D^{\beta_m} u \vee G^{\alpha_{m_1}} D^{\beta_{m_1}} u \quad \text{(do cálculo proposicional clássico)}$$

ou

$$x_l \leftrightarrow G^{\alpha_m} D^{\beta_m} x_m \vee y_l$$
$$x_l \leftrightarrow G^{\alpha_{m_1}} D^{\beta_{m_1}} x_{m_1} \vee y_l$$
$$\vdots$$
$$(3)\text{-}\quad x_l \leftrightarrow \left(G^{\alpha_m} D^{\beta_m} x_m \vee y_l\right) \vee \left(G^{\alpha_{m_1}} D^{\beta_{m_1}} x_{m_1} \vee y_l\right) \quad \text{(do cálculo proposicional clássico)}$$

Portanto, tem-se obrigatoriamente uma bifurcação causada pela disjunção inferida na fórmula 2 (ou 3) o que implica em dois ou mais ramos contrariando a hipótese inicial de um único ramo e mais de um vértice de entrada conectado à x_l, provando-se assim, por absurdo, a tese. ▲

Teorema 4.3

Num ramo terminado do tableau NK qualquer vértice onde ocorre a terminação de uma variável está conectado ao vértice onde ocorre a terminação da variável de entrada.

A demonstração será feita em duas etapas:

a) Um vértice onde ocorre a terminação de uma variável não pode ser auto conectado.

b) Um vértice de terminação de uma variável x não pode ser auto-conectado, conclui-se que ele só pode ser conectado à entrada ou a uma outra variável x' no mundo (k, l).

Prova de a) :

Se um vértice (i, j) da variável x é auto conectado, então existe um mundo (k, l) tal que:

$$T\,x\ \ (i, j)$$
$$(4)\text{-}\quad T\,x\ \ (k, l)$$

Onde

$$\left\{ \begin{array}{l} i - k = p > 0 \\ j - l = q > 0 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} i > k \\ j > l \end{array} \right\}$$

Mas, se (i,j) é um vértice de terminação da variável x, então: F x (i–1, j–β) $\forall\beta > 0$. Logo F x (i–1, j–q) mas, pela tautologia Gx → x obtem-se F Gx(i–1, j–l) → F x (i–2, j–l). Prosseguindo a iteração tem-se F x (i–p, j–q) ou F x (k, l) o que contradiz 4 . Portanto um vértice de terminação não pode ser autoconectado. ▲

Prova de b) :

- x' não pode ser auto-conectado a x, pois, pela transitividade da relação, x seria conectado a x (autoconectado).
- Se x' for autoconectada, tem-se que o número de vértices autoconectados deve ser finito pois senão x' não seria terminada, o que contraria a hipótese. Seja, então, (k', l') o menor par tal que x' é auto conectada em (k', l').
- Portanto, o vértice (k', l') da variável x' é conectado à variável de entrada u ou a alguma outra variável x" no vértice (p, q).
- A mesma análise pode ser aplicada à variável x" observando-se a exclusão das variáveis das etapas anteriores.
- Como o número de transições do sistema é finito, após um certo número r de iterações concluir-se-á que a variável x é conectada à variável u. ▲

Teorema 4.4

Num ramo terminado de um tableau NK, a posição relativa de todo vértice em relação ao vértice da variável de entrada é univocamente determinada.

Demonstração:

Dado um ramo terminado do tableau NK, a distância de conexão entre um vértice de terminação de uma variável x e o vértice de terminação da variável de entrada u é única.

Pelo teorema anterior, tem-se que essa conexão existe. Sejam os vértices (i, j) da variável x e (i–p,j–q) da variável u (com p≥ 0, q ≥ 0), portanto, tem-se:

$$\left\{ \begin{array}{c} x \leftrightarrow G^{n_1} D^{m_1} x_1 \vee y_1 \\ x_2 \leftrightarrow G^{n_2} D^{m_2} x_3 \vee y_2 \\ \vdots \\ x_{r-1} \leftrightarrow G^{r} D^{r} u \vee y_r \end{array} \right\} \rightarrow x \leftrightarrow G^{p} D^{q} u \vee z$$

Onde: $p = \sum_{i=1}^{r} n_i$; $q = \sum_{i=1}^{r} m_i$ e $z = G^{p-n_r} D^{q-m_r} y_r \vee \ldots \vee G^{n_1+n_2} D^{n_1+n_2} y_3 \vee G^{m_1} D^{m_1} y_2 \vee y_1$.

Se existirem outras possibilidades de conexão entre x e u através de z (outros "caminhos" no grafo a eventos) a fórmula acima pode ser escrita como: $x \leftrightarrow G^{p} D^{q} u \vee G^{p'} D^{q'} u \vee z'$.

Vai-se demonstrar que:

a) Se $p < p'$ então a distância (p, q) é única.

b) Qualquer outro vértice tem sua posição univocamente determinada em relação ao vértice de terminação da variável u.

Prova de a) :

Por absurdo, suponha que existe um ramo terminado tal que x é conectado à entrada u com distância de conexão (p', q'),

$$\left\{ \begin{array}{c} \left\{ \begin{array}{c} T\, x \\ T\, G^{p'} D^{q'} u \end{array} \right\} (i, j) \\ (5)\text{-}\ T\, u\,(i - p', j - q') \end{array} \right\}$$

onde: (i, j) é o vértice de terminação de x e (i–p',j–q') o vértice de terminação de u. Como a variável x termina em (i, j), tem-se F x (i–α, j–β) $\forall \alpha > 0$, $\forall \beta$. Como $p < p'$, tem-se $p' - p > 0$. Daí tem-se F (i–(p'+p), j–(q'–q)), ou F x (i–p'–p, j–q'+q). De $x \leftrightarrow G^{p} D^{q} u \vee G^{p'} D^{q'} u \vee z'$ tem-se então:

$$F G^{p} D^{q} u (i - p' + p, j - q' + q)$$

$$\downarrow$$

$$(6)\text{-}\ Fu\,(i - p', j - q')$$

Contradição entre 5 e 6. Provando, por absurdo, a tese proposta. ▲

Prova de b):

Suponha que a variável x tem um vértice no mundo (i, j), conectado a um vértice da variável x_1 no

mundo $(i-n_1, j-m_1)$, isto é:

$(x \leftrightarrow G^{n_1}D^{m_1}x_1 \vee G^{n_2}D^{m_2}x_2 \vee \ldots \vee G^{n_p}D^{m_p}x_p \in \Omega)$, gerando, então a seguinte situação:

$$\left\{\begin{array}{c} T\,x \\ T\,G^{n_1}D^{m_1}x_1 \\ \vdots \\ T\,G^{n_k}D^{m_k}x_k \\ F\,G^{n_{k+1}}D^{m_{k+1}}x_{k+1} \\ \vdots \\ F\,G^{n_p}D^{m_p}x_p \end{array}\right\}(i,j)$$

$$T\,x_1(i-n_1, j-m_1)$$
$$\vdots$$
$$T\,x_k(i-n_k, j-m_k)$$
$$F\,x_{k+1}(i-n_{k+1}, j-m_{k+1})$$
$$\vdots$$
$$F\,x_p(i-n_p, j-m_p)$$

As variáveis $x_1 \ldots x_k$ tem necessariamente vértices nos mundos $(i-n_1, j-m_1)$ ... $(i-n_k, j-m_k)$, pois tem-se $F\,x(i-1, j) \Rightarrow FG^nD^m\,x'(i-1, j) \Rightarrow Fx'$

$(i-n-1, j-m)$ e $F\,x(i, j+1) \Rightarrow F\,G^nD^mx'(i, j+1) \Rightarrow F\,x'(i-n, j-m+1)$.

Supondo-se, por absurdo, que haja um ramo terminado onde:

1- $Tx(i-1,j)$

ou

2- $Tx(i, j+1)$

ou

3- $Fx(i, j)$

Pois, dessa forma, o vértice de x em (i, j) seria diferente.

Se tomarmos como tendo ocorrido a situação 1, tem-se como conseqüências:

$$Tx_1(i - n_1 - 1, j - m_1)$$
ou
$$\vdots$$
ou
$$Tx_p(i - n_p - 1, j - m_p)$$

Logo, o vértice de alguma variável $x_1 ... x_p$ foi alterado, pois no ramo original tinha-se $F\ x_1 \ldots F\ x_p$.

Se ocorrer a situação 2, ter-se-á:

$$Tx_1(i - n_1, j - m_1 + 1)$$
ou
$$\vdots$$
ou
$$Tx_p(i - n_p, j - m_p + 1)$$

implicando, como no caso anterior que o vértice de alguma variável $x_1 \ldots x_p$ foi alterado, pois no ramo original tinha-se $F\ x_1 \ldots F\ x_p$.

Ocorrendo a situação 3 tem-se

$$Fx_1(i - n_1, j - m_1)$$
$$\vdots$$
$$Fx_k(i - n_k, j - m_k)$$

Alterando, no ramo original, todos os vértices de x_1 a x_k.

Conseqüências:

- Alguma das variáveis x_1 a x_p terá um dos seus vértices alterado.
- Algum outro vértice de alguma outra variável será alterado.

Como todas as variáveis são terminadas alterar-se-á a posição relativa de um vértice de terminação, o que não é possível dado a demonstração feita em b, provando por absurdo a tese proposta. ▲

Teorema 4.5

Um ramo é terminado no tableau NK, se e somente se ele corresponde a uma solução causal para as equações do conjunto Ω.

Demonstração:

Da definição 6 temos que, para que um elemento h seja causal ele deve ter os vértices dos cones a sudeste que o compõem no quadrante nordeste do plano n×t (acima do eixo-n e a direita do eixo-t). Pela correspondência entre o plano n×t e os mundos possíveis do conjunto W essa situação se verifica se os vértices de cada variável do ramo terminado se encontrarem à direita e acima do vértice da variável de entrada u. Ou seja, para toda variável proposicional p com vértice em (i, j) o vértice da variável u deve estar em (i–k, j–l) com $k \geq 0$ e $l \geq 0$. Pelo teorema 4.1 temos que todo vértice de toda variável, exceto a variável de entrada está conectado a algum vértice de alguma variável e pelo teorema 4.2 temos que num ramo terminado do tableau NK **todo vértice** onde ocorre a terminação de uma variável está conectado ao vértice onde ocorre a terminação da variável de entrada. Considerando que a distância de conexão é sempre positiva, conclui-se que a terminação de qualquer variável proposicional ocorre num ponto a nordeste da terminação da variável de entrada. Logo, em todo o ramo terminado teremos todos os vértices das variáveis proposicionais do ramo conectadas ao vértice da variável de entrada u (figura 4.9), satisfazendo a condição de causalidade. A recíproca é trivialmente verdadeira, visto que se a solução é causal, então seu modelo NK correspondente é dado por um ramo terminado..▲

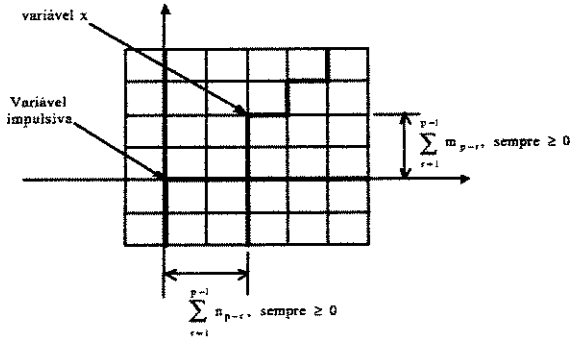


figura 4.9

Vértices de variáveis conectados à variável de entrada u

Teorema 4.6

Um ramo terminado do tableau NK corresponde a uma solução causal única (e portanto mínima) para as equações do conjunto Ω.

Demonstração:

Sabemos pela completude do tableau que se há uma solução para o conjunto de equações de Ω, então existe um ramo aberto correspondente a ela no tableau. A univocidade foi demonstrada no teorema 4.4 que enuncia "num ramo terminado de um tableau NK, a posição relativa de todo vértice em relação ao vértice da variável de entrada é univocamente determinada", mas o dióide sempre apresentará, entre quaisquer soluções, a solução mínima conforme o teorema 9-ii de Cohen., et al (1989). Portanto, como nesse caso ela é única, temos que também é mínima. ▲

Teorema 4.7

Se existe um modelo não impulsivo então existe um modelo impulsivo.

Demonstração:

Da isotonia do dióide M em relação ao produto (definição 2.8) tem-se que:

Se $y \leqslant z$ então $yu \leqslant zu$, ou seja, por contraposição: Se $\neg(yu \leqslant zu)$ então $\neg(y \leqslant z)$. Portanto, se existe um tableau que verifica um modelo para F $yu \rightarrow zu$ então F $y \rightarrow z$ é verificável. Pela completude da lógica NK conclui-se que o tableau existe. ▲

Considerações Acerca dos Resultados Anteriores.

Na introdução a este capítulo foi apresentado um exemplo de verificação de uma especificação para um GET onde o tableau retorna um ramo associado a uma solução não-mínima e não-causal, duas características não desejadas para uma solução, pois deseja-se encontrar uma solução mínima, causal e impulsiva. Com o objetivo de dotar o tableau de um mecanismo que o permita retornar apenas ramos associados a essas características desejadas foram demonstrados sete teoremas que permitem ao tableau identificar um ramo associado a uma solução mínima, causal e impulsiva pela identificação de uma característica denominada terminação, ou seja, se o ramo é terminado então corresponde à solução desejada. Já os quatro primeiros teoremas demonstram propriedades relativas aos vértices de um ramo terminado necessárias para as três últimas demonstrações. Baseado nas demonstrações 4.5, 4.6 e 4.7, é proposto no ítem 4.4 abaixo um algoritmo que permite identificar quando um ramo do tableau correspondente a solução mínima, causal e impulsiva. No passo 5 do algoritmo são introduzidos os parâmetros imax e jmax como critério de parada para a busca de um ramo terminado pelo tableau. O parâmetro jmax indica o máximo número de aplicações da regra π a partir de qualquer mundo possível (i–k, j) com $0 \leq k \leq$ imax, o o parâmetro imax indica o máximo número de aplicações da regra ν a partir de qualquer mundo possível (i, j–m) com $0 \leq m \leq$ jmax, sendo imax e

jmax definidos *a priori* e dependendo das características particulares do sistema no qual se esteja verificando a especificação. Pode-se garantir que os parâmetros imax e jmax funcionam como critério de parada para o algoritmo verificando-se que a sua definição antes do início da execução do algoritmo torna finito o número de nós empilhados nas pilhas 1 e 2 devido às bifurcações geradas pela aplicação da regra β, pois como o número de fórmulas num mundo possível é finito, como o número de disjuntos numa fórmula é finito, e as regras tableau podem ser aplicadas apenas dentro dos limites definidos por imax e jmax tem-se que os nós empilhados serão em número finito. No momento em que todos os nós empilhados forem utilizados pelo algoritmo na criação de novos ramos as pilhas ficarão vazias garantindo a parada do algoritmo. Na implementação que será proposta no próximo capítulo imax e jmax deverão ser definidos pelo usuário inserindo-se os valores correspondentes a ambos antes do início da verificação de uma especificação pelo tableau. Ressalta-se entretanto que caso o tableau não encontre um ramo terminado dentro dos limites fixados pelos parâmetros imax e jmax não significa que este não possa ser encontrado num espaço de busca onde imax e jmax possuam valores maiores do que os utilizados na busca.

4.4 Algoritmo.

Os resultados anteriores garantem que somente os ramos abertos e impulsivamente terminados são interessantes para se encontrar as soluções esperadas via tableau, os outros podem ser descartados (conforme ilustra o esquema de decisão do algoritmo na figura 4.10 abaixo).

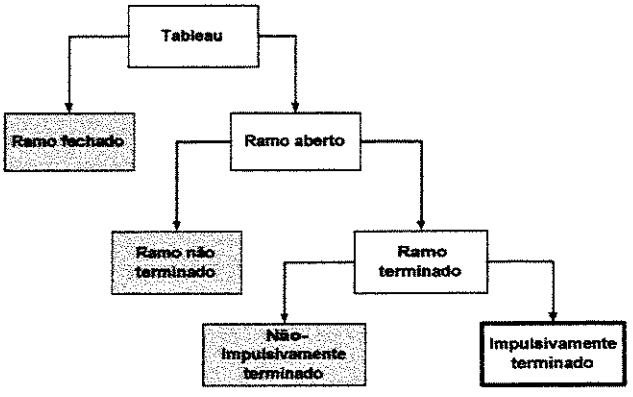


figura 4.10

Esquema para o algoritmo tableau

O algoritmo proposto pressupõe os seguintes dados:

a) Um conjunto de NK-fórmulas associadas ao GET como discutido no capítulo 3.

b) Uma especificação do tipo $y \rightarrow z$.

c) Limites máximos de busca dentro de um mesmo ramo, expressos pelos inteiros imax e jmax.

Durante a execução do algoritmo haverá sempre um "ramo do tableau corrente" que será desenvolvido a partir da visita a mundos possíveis seguindo o padrão especificado na figura 4.11 abaixo.

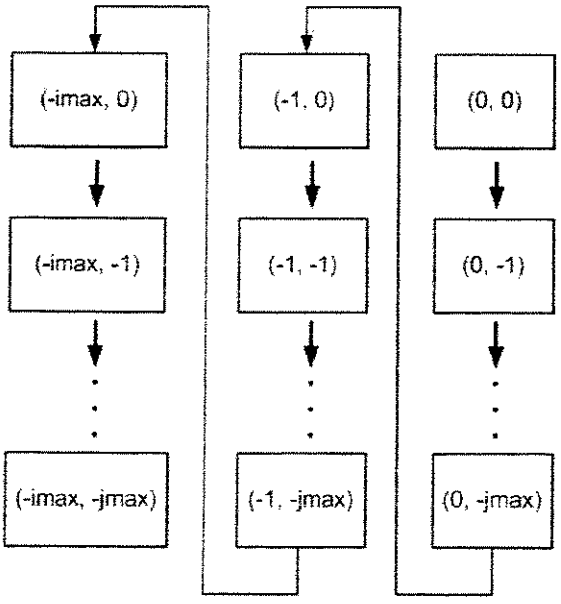


figura 4.11

Seqüência de abertura de mundos possíveis num ramo tableau

Durante este desenvolvimento, as ocorrências de fórmulas tipo β (e portanto de bifurcações no tableau) deverão ser empilhadas. Caso o "ramo corrente" não constitua um ramo terminado as bifurcações deverão ser recuperadas, dando orígem a novos "ramos correntes".

Deve-se observar que quando uma variável proposicional não tem valor-verdade definido pelas regras tableau, há implicitamente uma bifurcação (conseqüência da lei do 3° excluído). Estas bifurcações são acumuladas numa segunda pilha.

Um ramo será considerado descartável se ocorrer uma das situações ilustradas pelos blocos em cinza da figura 4.10, isto é, se o ramo for fechado, se o ramo não for terminado, ou se o ramo não for impulsivamente terminado.

Algoritmo:

1- **Se** o algoritmo estiver sendo iniciado **então**

insira F $y \rightarrow z$ (iniciando o 1° ramo)

Senão se as pilhas estiverem vazias **então** pare

senão inicie um novo ramo a partir da pilha 1 (se a pilha 1 estiver vazia inicie da pilha 2)

2- Aplique as regras ν e π a partir de todos os mundos precedentemente visitados no presente ramo.

3- Escreva todas as α-fórmulas e seus respectivos componentes.

4- Escreva todas as β-fórmulas (um dos disjuntos) e empilhe (pilha 1) os disjuntos que darão origem aos outros ramos a serem abertos posteriormente;

5- Aplique a lei do terceiro-excluído para todas as variáveis proposicionais e empilhe as bifurcações (pilha2);

6- **Se** o ramo é *terminado* **então** retorne o ramo e pare;

7- **Se** o ramo for *descartável*, **então** vá para o passo 1;

8- **Se** a dimensão máxima (imax, jmax) foi atingida **então** vá para o passo 1

Senão altere o mundo (conforme a figura 4.11) e vá para o passo 2;

4.5 Conclusão

Neste capítulo foram apresentados conceitos como conectividade e terminalidade e teoremas ligando esses conceitos a propriedades como a de minimalidade, causalidade, unicidade e impulsividade de uma solução obtida via tableau. Baseado nesses teoremas foi proposto um algoritmo capaz de retornar uma única solução causal, mínima e impulsiva ao se encontrar um ramo terminado. No capítulo 5 tratar-se-á da implementação computacional deste algoritmo e no capítulo 6 apresentar-se-ão exemplos de verificações de especificações via tableau para alguns GET's.

CAPÍTULO 5.

Implementação Computacional do Algoritmo Tableau.

5.1 Introdução.

Neste capítulo será apresentada uma descrição da implementação computacional do algoritmo descrito no fim do capítulo anterior. Como a linguagem de programação utilizada é uma linguagem orientada a objetos, inicialmente serão descritas as principais classes da implementação e, portanto, dos objetos criados para armazenar os dados necessários ao programa. Em seguida a seqüência de instanciação dos objetos pelo programa será descrita através de fluxogramas.

5.2 Linguagem de Programação.

A linguagem de programação escolhida para a implementação do algoritmo tableau foi a linguagem Java™, uma linguagem de programação de alto nível, orientada a objetos, robusta e multiplataforma Vários foram os motivos para a adoção de uma linguagem como Java™, entre eles pode-se mencionar a maior clareza do código, a ausência da aritmética de ponteiros (cujo uso pode dificultar o reaproveitamento do código já criado ou possíveis correções) a "reusabilidade" do código facilitando modificações e aperfeiçoamentos no projeto enquanto o aperfeiçoamento da máquina virtual Java™ (JVM) fornece continuamente a aportunidade de melhoria da eficiência na execução do código.

5.3 Classes Para Implementação do algoritmo Tableau.

As classes do programa foram criadas de modo a ter como menor elemento objetos que representem fórmulas do tipo G^nD^mx (classe AtomicNo), passando a seguir por fórmulas do tipo $x_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$ (classe NoMega) que podem ser representadas por elementos da classe AtomicNo (o disjunto $\bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$ pode ser representado por um array de objetos AtomicNo).

5.3.1 Classe AtomicNo.

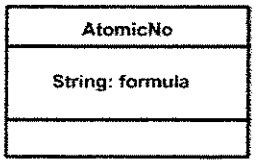


figura 5.1
Diagrama da classe AtomicNo

Para armazenar o que se convencionou como a menor unidade de informação relativa às fórmulas inseridas e manipuladas no programa criou-se uma classe chamada AtomicNo (que não corresponde necessariamente a um átomo conforme definido no capítulo de apresentação da lógica NK). Esta classe permite representar fórmulas do tipo G^nD^mx através de um Objeto String (que é uma instância da classe Java™ String equivalente a um array de caracteres) e dois tipos primitivos inteiros que armazenam a variável x, *n* (o expoente de G) e *m* (o expoente de D) respectivamente.

Como exemplos podem-se tomar as fórmulas G^2D^4y, GD^3x1, G^5z e D^3y conforme mostra a tabela 5.1 abaixo:

fórmulas do tipo G^nD^mx	Variável	expG	expD
G^2D^4y	y	2	4
GD^3x1	x1	1	3
G^5z	z	5	-1
D^3y	y	-1	3

Tabela 5.1
Exemplos de como a classe AtomicNo armazena fórmulas do tipo G^nD^mx

5.3.2 Classe NoMega.

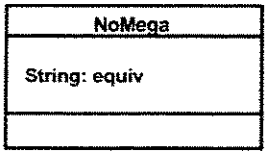


figura 5.2
Diagrama da classe NoMega

Fórmulas que possuem um operador binário podem ser representadas através de um array de Objetos com um "AtomicNo" em cada célula deste array (no atual estado da implementação, a ordem

dos objetos AtomicNo no array é a mesma dos seus equivalentes inseridos pelo usuário no programa). Representa-se através das instâncias dessa classe as equivalências do conjunto Ω advindas do grafo a eventos, essas fórmulas possuem, conforme visto no capítulo 3, a forma:

$$x_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$$

Para representar completamente fórmulas deste tipo as instâncias da classe NoMega conterão três elementos (tabela 5.2 abaixo) representando a variável x_i à esquerda da bi-implicação como uma instância da classe AtomicNo; o disjunto da direita através de um array de "AtomicNo"; e um terceiro elemento que é criado automaticamente na instanciação da classe NoMega e que indica aos métodos do programa a dimensão do array de "AtomicNo".

fórmulas do tipo $x_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$	equi1	equi2	size
$x1 \leftrightarrow G^3 Dx2 \vee Gx3 \vee x4$	x1, -1,-1	x2, 3, 1 x3, 1, -1 x4, -1,-1	3
$x1 \leftrightarrow Dx2 \vee GDx3 \vee G^3D^3x4 \vee G^5x5$	x1, -1, -1	x2, -1, 1 x3, 1, 1 x4, 3, 3 x5, 5, -1	4

tabela 5.2

Exemplos de representação de fórmulas do tipo $x_i \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$ pela classe NoMega

5.3.3 Classe Omega

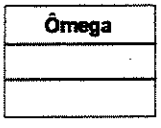


figura 5.3

Diagrama da classe Omega

As instâncias da classe NoMega serão armazenadas em uma instância da classe Omega. A classe Omega estende a classe arrayList do pacote "util", um dos pacotes padrão da linguagem Java™. A classe ArrayList fornece um vetor para armazenamento de objetos, possuindo dimensão auto ajustável e com métodos de busca pré-definidos. Esses métodos padrão foram sobrescritos pelos métodos da

classe Omega permitindo assim que as buscas realizadas num objeto "omega" sejam feitas de forma rápida e com métodos reduzidos, uma característica necessária para um objeto que deverá ser percorrido sempre que um novo "No" for inserido num mundo possível.

5.3.4 Classe No.

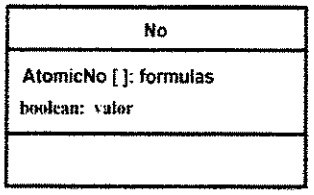


figura 5.4
Diagrama da classe No

Todos os nós de um mundo possível podem ser armazenados na memória na forma de instâncias da classe No. Os argumentos utilizados para sua criação são um array de "AtomicNo" representando uma fórmula do tipo $\bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$ e um booleano que indica o seu assinalamento ("T" ou "F"). Na tabela 5.3 abaixo encontram-se exemplos de objetos "No" representando as fórmulas: $G^3Dx2 \vee Gx3 \vee x4$; $Dx2 \vee GDx3 \vee G^3D^3x4 \vee G^5x5$; GDx1; Gx3 e x6. Os booleanos *usedPi, usedNi* indicam ao programa se o "No" já sofreu a aplicação da regra π ou da regra ν respectivamente. O booleano *usedOm* indica se o "No" já foi utilizado para a inserção de fórmulas equivalentes advindas do conjunto Ω. O booleano *usedAB* indica se o "No" sofreu a aplicação das regras α ou β . O inteiro *fsize* indica ao programa a dimensão do array de "AtomicNo". Com estes elementos o programa pode armazenar e manipular de maneira adequada as fórmulas contidas nos mundos possíveis de um "ramo" do tableau.

	formulas (AtomicNo[])	fsize	valor	usedAB	usedOm	usedNi	usedPi
$G^3Dx2 \vee Gx3 \vee x4$	x2, 3, 1 x3, 1, -1 x4, -1,-1	3	true	true	false	false	false
$Dx2 \vee GDx3 \vee G^3D^3x4 \vee G^5x5$	x2, -1, 1 x3, 1, 1 x4, 3, 3 x5, 5, -1	3	false	true	false	false	false
GDx1	x1, 1, 1	1	true	false	false	true	true
Gx3	x3, 1, -1	1	false	false	false	true	false
x6	x6, -1, -1	1	true	false	true	false	false

tabela 5.3
Exemplos de como fórmulas disjuntivas, ou não, são armazenadas numa instância da classe NO

5.3.5 Classe Mundo

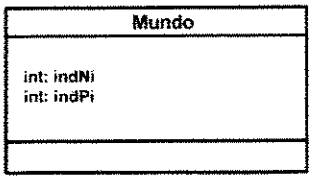


figura 5.5
Diagrama da classe Mundo

As instâncias da classe Mundo (extensão da classe ArrayList) são utilizadas para armazenar todos os objetos "No" com os mesmos índices "I" e "J", ou seja, todos as fórmulas de um dado mundo possível. O objeto mundo, ao ser criado, recebe um par de índices (veja figura abaixo) que o posiciona em relação ao mundo (i–0,j–0) e que, quando da aplicação da regra β (ou equivalentemente, da lei do 3°excluído) a uma fórmula de um "No", serve para indicar o mundo de onde partirá a bifurcação gerada pela aplicação desta regra conforme será mostrado adiante.

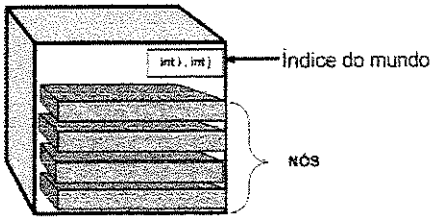


figura 5.6
Representação dos elementos da classe Mundo

5.3.6 Classe Pilha

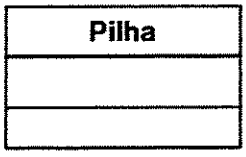


figura 5.7
Diagrama da classe Pilha

A classe pilha instancia uma classe da biblioteca "util", a classe Stack, que possui as propriedades de uma pilha (o primeiro objeto que entra é o último que sai). A classe Stack possui métodos próprios que facilitam a busca de objetos empilhados, sendo esses métodos sobrescritos pelos métodos da

classe Pilha. Pelo procedimento de empilhamento os primeiros disjuntos abertos pela regra β (ou equivalentemente, da lei do 3°excluído) serão os últimos a serem verificados conforme ilustra a figura abaixo. A classe Pilha será instanciada para a criação das pilhas 1 e 2 descritas no algoritmo apresentado ao fim do capítulo 4.

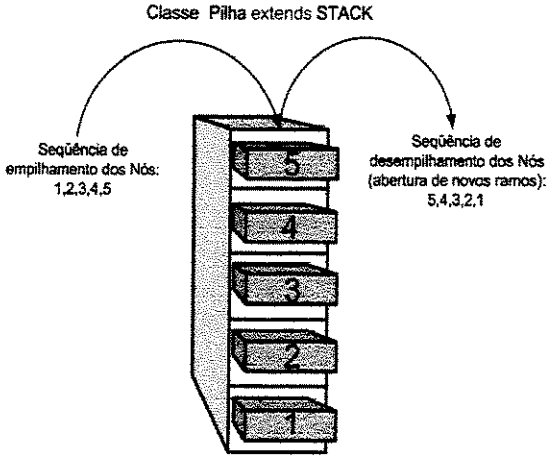


figura 5.8
Representação da ordem de empilhamento num objeto Pilha

5.3.7 Classe NoEmp

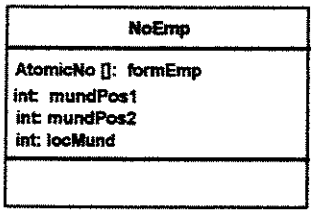


figura 5.9
Diagrama da classe NoEmp

A informação referente aos "Nos" empilhados pela aplicação da regra β (ou equivalentemente, da lei do 3°excluído) serão armazenadas na forma de instâncias da classe NoEmp, que constará de um

array de "AtomicNo" que armazenará uma fórmula do tipo $\bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} x_j$, um índice *mundPos1* referente à coluna do "Ramo" onde se aplicou a regra, um índice *indPos2* referente à célula da coluna onde se aplicou a regra e um índice *locMund* referente à célula do objeto mundo onde se aplicou regra. Com estes indicativos torna-se possível ao programa recolocar o "No" empilhado pela aplicação de uma regra β (ou equivalentemente, da lei do 3°excluído) no seu lugar original no "Mundo", para que a bifurcação do "Ramo", que se inicia com este "No", possa ser visitada.

Uma vez descritos os objetos auxiliares para a execução da regra β (ou equivalentemente, da lei do 3°excluído) torna-se clara a ilustração abaixo, onde vê-se um dos disjuntos inserido no mundo possível enquanto o outro disjunto é inserido na pilha onde ficará à disposição dos métodos do programa para uma possível inserção no "Ramo".

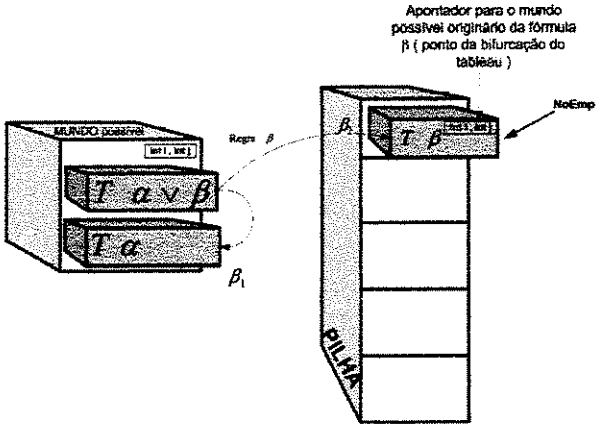


figura 5.10

Representação da aplicação da regra β na fórmula $\alpha \vee \beta$

5.3.8 Classe Ramo.

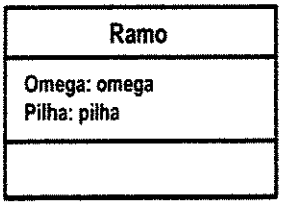


figura 5.11

Diagrama da classe "ramo"

A aplicação da regra ν num mundo em que o índice indPi = 0 implica na criação de uma nova coluna do "Ramo" tornando necessária a instanciação das classes Coluna e Mundo. Torna-se necessário armazenar estas instâncias da classe coluna na mesma ordem em que são criadas. A classe Ramo, que estende as propriedades da classe ArrayList, será utilizada para armazenar as instâncias da classe Coluna (figura 5.12). A dimensão máxima de um "Ramo" é dada por dois índices definidos pelo usuário: O índice imax define o número máximo de "Colunas" e o índice jmax define o número máximo de "Mundos" em uma "Coluna".

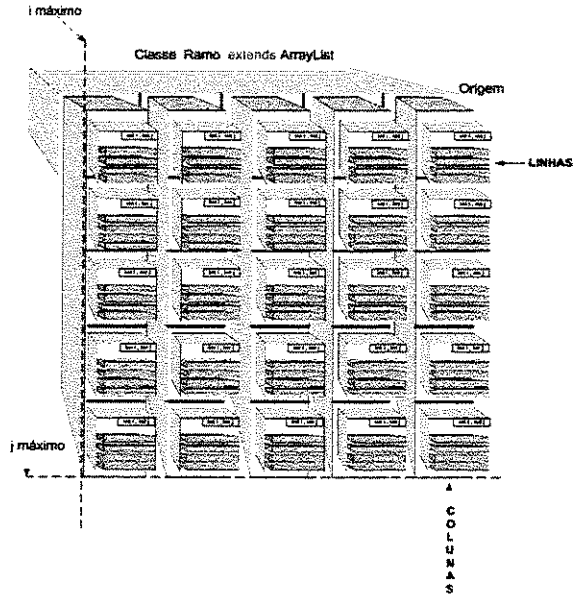


figura 5.12
Representação de um objeto do tipo Ramo

A figura 5.13 abaixo ilustra a instanciação de um "mundo" possível de índices (indNi – i, indPi – j pela aplicação da regra ν ao "mundo" de índices (indNi – i, indPi – 1) e da regra π ao "mundo" de índices (indNi – 1, indPi – j). Se o "mundo" que estiver sendo instanciado tiver índices (0, indPi – j) então apenas o "mundo" de índice (0, indPi – j + 1) contribuirá com fórmulas sofrendo a aplicação da regra π , pois trata-se da primeira "coluna" do "ramo". Se o "mundo" que estiver sendo instanciado tiver índices (indNi – i, 0) então apenas o "mundo" de índice (indNi – i + 1, 0) contribuirá com fórmulas sofrendo a aplicação da regra ν , pois trata-se da primeira "linha" do "ramo". O "mundo" de índices (0, 0) recebe apenas a especificação e a aplicação da regra α e da regra β, pois é o primeiro "mundo" do "ramo".

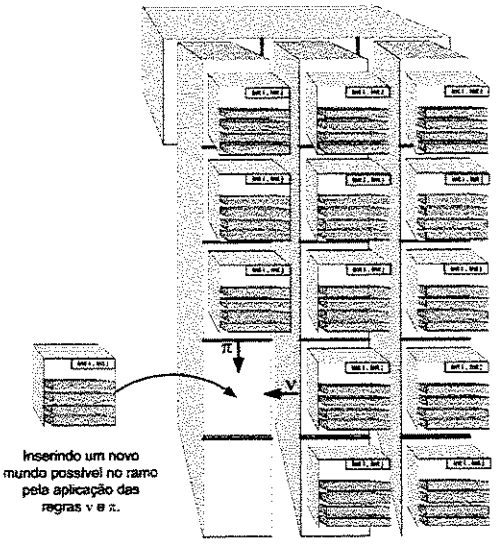


figura 5.13

Representação da criação de um objeto do tipo Mundo

5.4 Implementação das Regras Tableau

5.4.1 Regras α e β.

As regras α e β distinguem-se pelo fato de a regra α se restringir ao mesmo mundo da disjunção assinalada por F que sofre a aplicação da regra, enquanto a regra β, por originar uma bifurcação exige a utilização de uma pilha para armazenar um dos componentes do disjunto que originará a bifurcação através de uma instância da classe NoEmp.

A regra α se encontra ilustrada na figura abaixo e consiste na instanciação de dois novos Nos que possuem como fórmulas os elementos do disjunto onde se está aplicando a regra α mantendo-se o mesmo assinalamento (no caso F) nos "Nos" gerados.

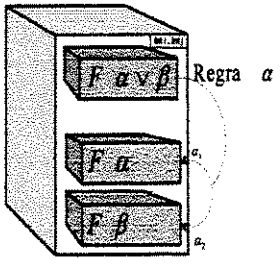


figura 5.14

Representação da aplicação da regra α

A regra β foi ilustrada quando da apresentação das classes Pilha e NoEmp, que fornecem os objetos necessários para armazenar a informação referente ao disjunto que foi empilhado e que será utilizado na abertura de um novo "ramo" (figura 5.10), caso o "ramo" atualmente verificado pelo tableau seja descartável conforme definido no capítulo 4.

5.4.2 Regras ν e π.

As regras ν e π causam a inserção do nó resultante de sua aplicação num outro "mundo" possível contíguo ao "mundo" onde se encontra o nó que sofre a aplicação da regra.

A aplicação da regra ν se encontra ilustrada na figura abaixo, no caso do mundo possível contíguo ainda não existir será criada uma instância da classe Coluna e uma instância da classe Mundo inserindo-se nesse "mundo" recém instanciado uma instância da classe No contendo a fórmula que sofreu a aplicação da regra ν, com o mesmo assinalamento e com o seu expoente em "G" acrescido de uma unidade.

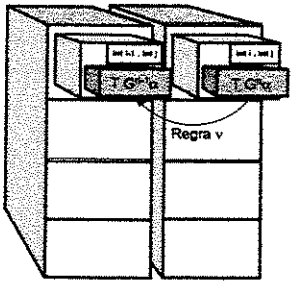


figura 5.15
Representação da aplicação da regra ν

A aplicação da regra π não implica na instanciação de uma "coluna", mas apenas de um "mundo" possível, que conterá, por sua vez, uma instância da classe No contendo a fórmula que sofreu a aplicação da regra π, com o mesmo assinalamento e com o seu expoente em "D" acrescido de uma unidade conforme mostra a figura abaixo.

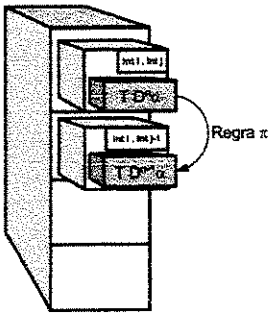


figura 5.16

Representação da aplicação da regra π

5.5 Dinâmica de Instanciação das Classes do Programa Tableau.

De maneira geral o procedimento para verificação de especificações no programa tableau é mostrado na figura abaixo.

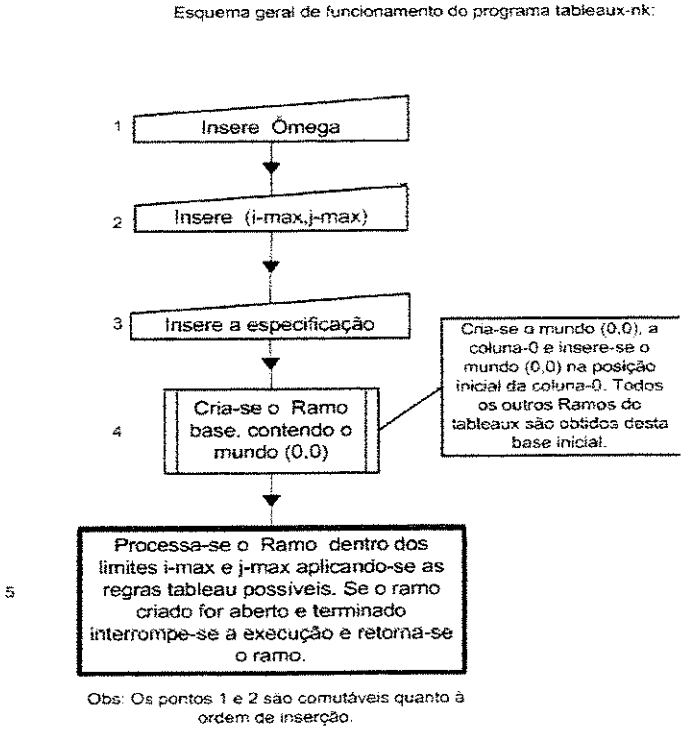


figura 5.17

Procedimento geral para o funcionamento do tableau-NK

O Detalhamento da instanciação de objetos durante a execução da implementação pode ser representado pelo fluxograma da figura 5.18 abaixo, podendo-se verificar que, havendo a possibilidade

de se encontrar um "ramo" terminado que falseie a especificação a ser verificada dentro dos limites para a aplicação das regras ν e π dados por imax e jmax, então este "ramo" será encontrado. Conforme foi ressaltado no capítulo 4 os limites imax e jmax são fixados arbitrariamente antes da verificação da especificação ser iniciada, portanto, caso o programa não retorne um "ramo" aberto isto não significa que a especificação não possa ser falseada, podendo ocorrer o falseamento dentro de limites imax e jmax maiores do que os inicialmente determinados.

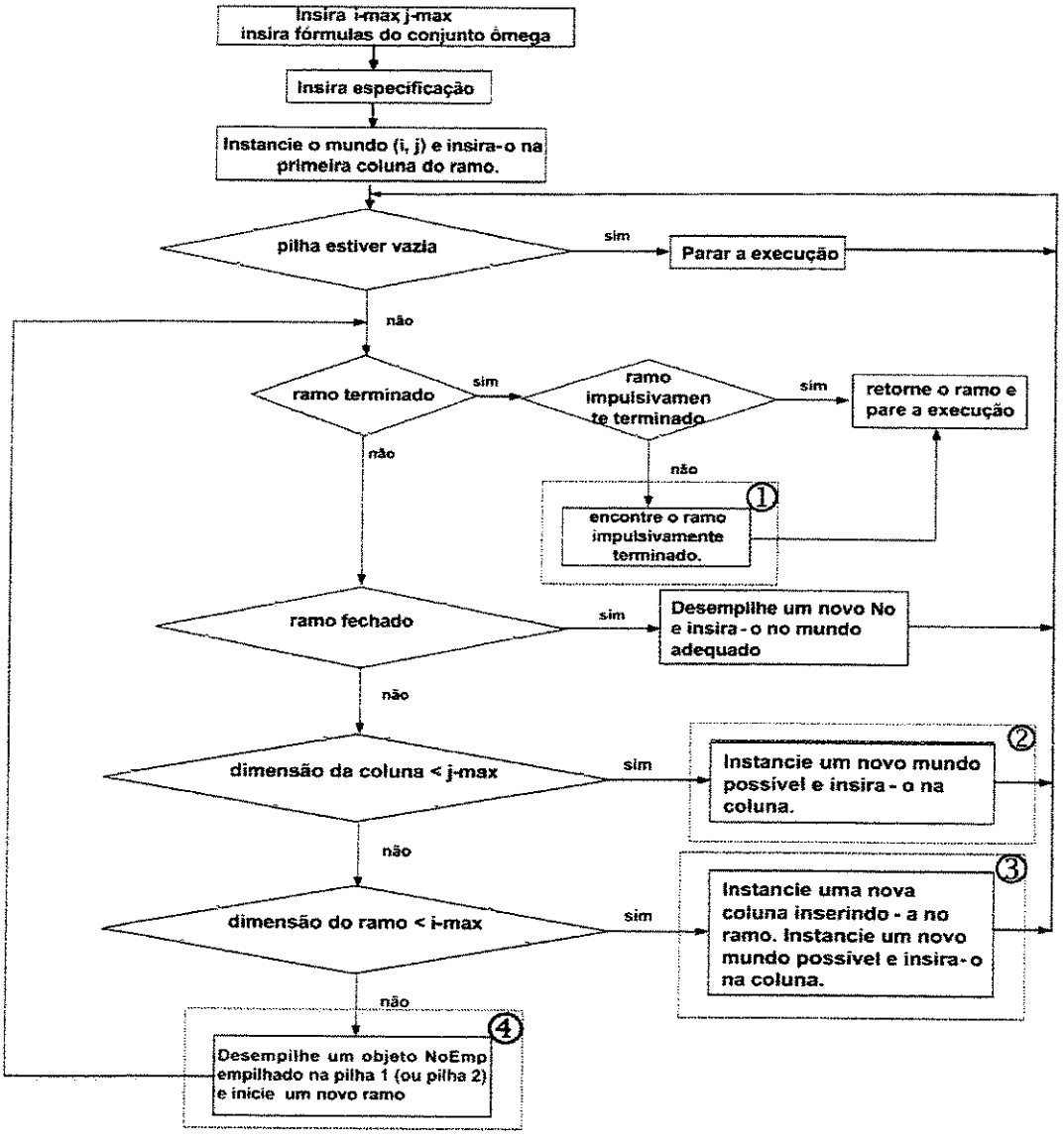


figura 5.18

Procedimento de decisão para a criação de mundos possíveis no algoritmo tableau

Nas figuras 5.19 a 5.22 são apresentados os detalhamentos das instruções 1,2,3 e 4 do fluxograma

da figura 5.18.

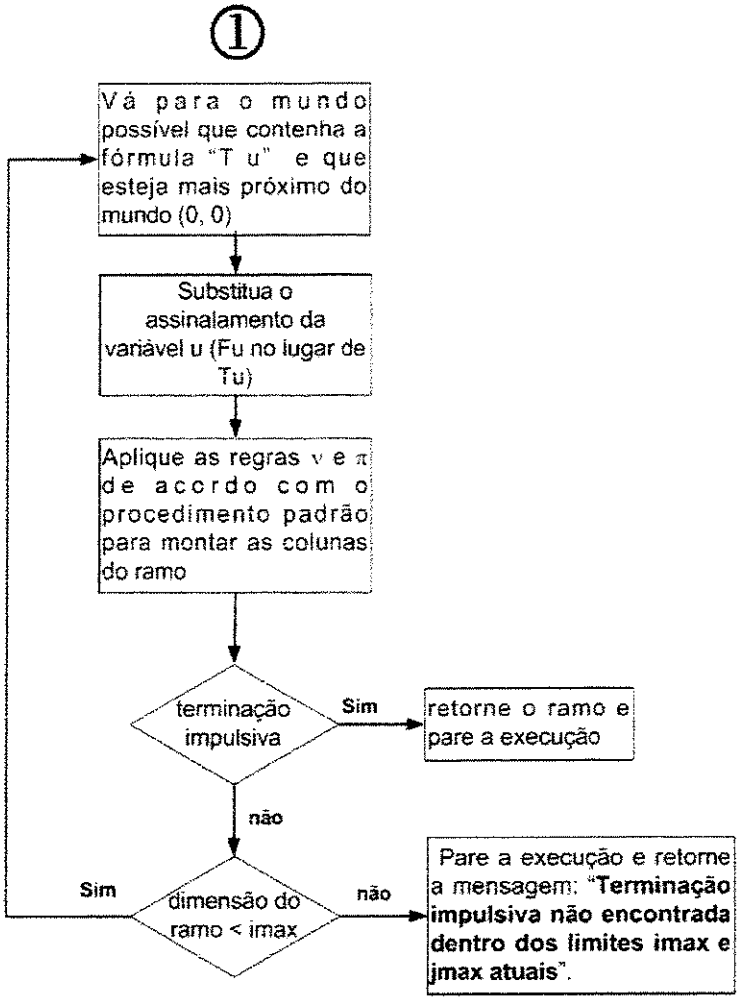


figura 5.19

Detalhamento da instrução 1 do fluxograma da figura 5.18

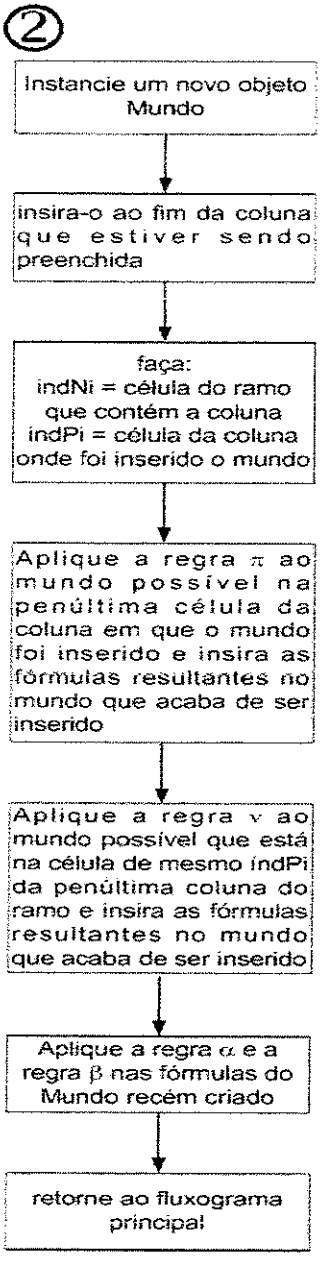


figura 5.20

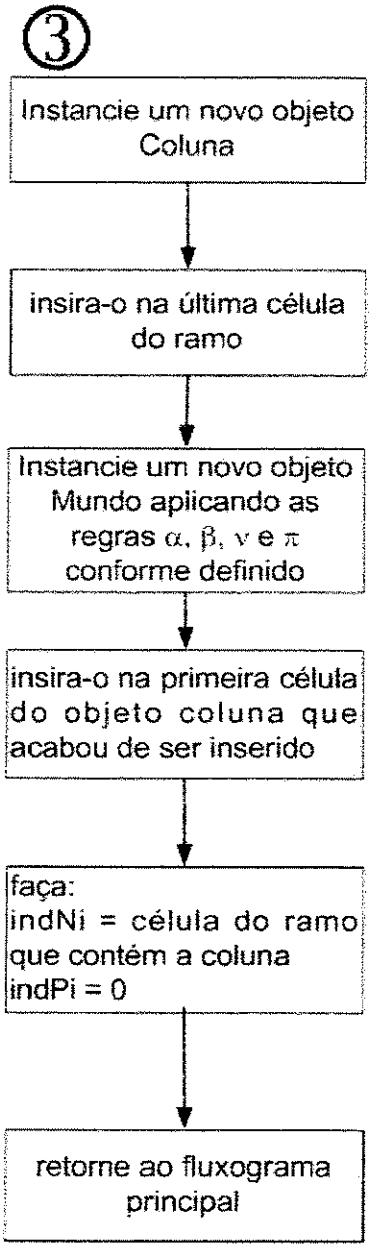


figura 5.21

Detalhamento das instruções 2 e 3 do fluxograma da figura 5.18

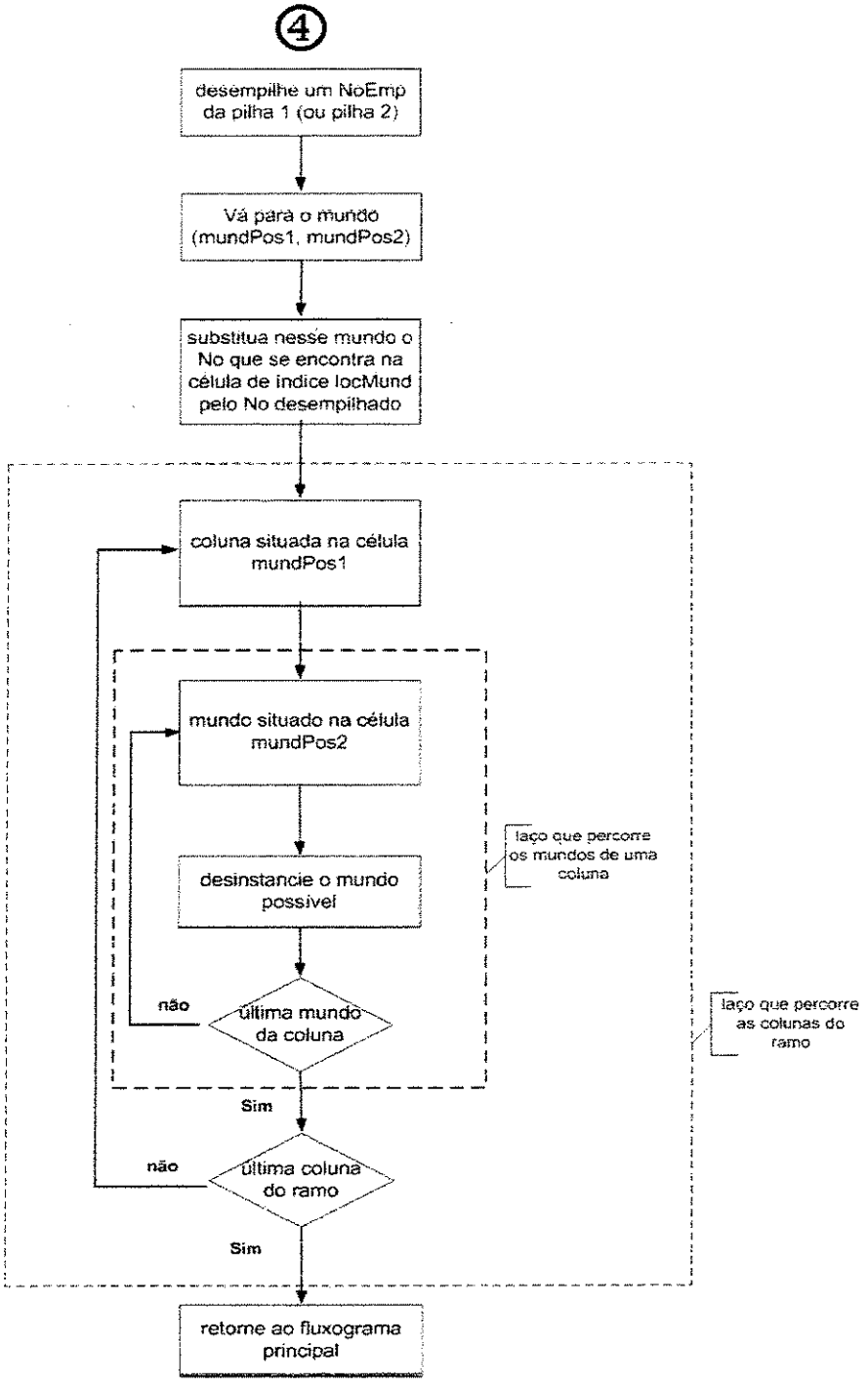


figura 5.22

Detalhamento da instrução 4 do fluxograma da figura 5.18

Conclusão

No presente capítulo foram apresentados detalhes da implementação computacional do algoritmo descrito ao fim do capítulo 4. Seguindo o paradigma da orientação a objetos da linguagem de programação utilizada na implementação, no caso a linguagem Java™, foram apresentados primeiro os detalhes das classes do programa passando-se a seguir para a implementação das regras tableau e, em seguida, via fluxogramas, apresenta-se a dinâmica da instanciação de objetos na execução da implementação. No próximo capítulo serão apresentados exemplos de verificação de especificações para diversos GET's utilizando-se o algoritmo proposto ao fim do capítulo 4.

Capítulo 6

Exemplos de Verificações de Especificações Tableau NK.

6.1 Introdução

Neste capítulo serão vistos exemplos de verificação de especificações em diferentes GET's utilizando-se o algoritmo tableau proposto no capítulo 4. Inicialmente serão mostrados os resultados de verificações de especificações em alguns grafos com uma entrada 'u', ao fim do capítulo o último exemplo tratará de um GET com mais de uma entrada para que se possa discutir como o tableau trata casos de GET's multi-entradas. A seqüência de figuras utilizadas na apresentação de cada exemplo será: GET onde se verifica a especificação, trajetórias no plano n×t das transições que se está comparando via tableau e ramo(s) aberto(s) retornado(s) pelo tableau.

6.2 Exemplos

Exemplo 1

Inicialmente será utilizado o grafo do capítulo 2 com saída $y = \delta^2(\gamma\delta)^*$.

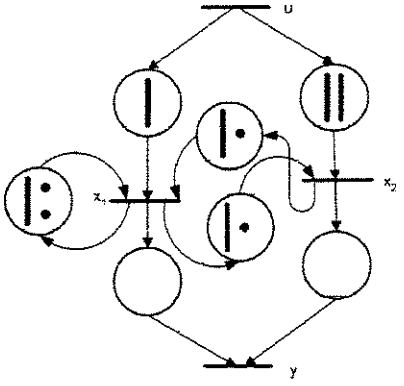


figura 6.1
GET para o exemplo 1

a trajetória da saída y pode ser vista no plano n×t abaixo:

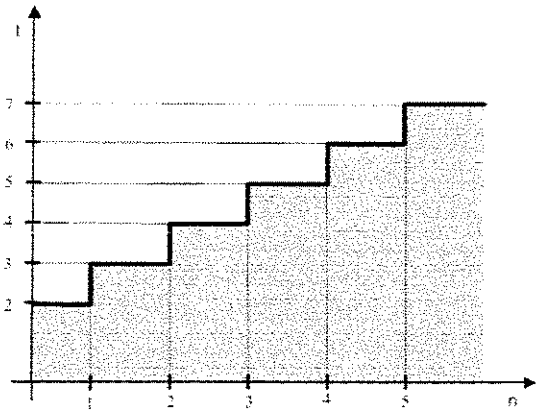


figura 6.2
Saída do GET da figura 6.1 no plano

Nesse exemplo deseja-se verificar se 4 unidades de tempo após uma entrada impulsiva, haverá algum disparo na saída, isto é, deseja-se verificar se a saída y é sempre menor que δ^4. As trajetórias de y e δ^4 são mostradas na figura 6.3. Esta especificação traduzida na lógica NK resulta na fórmula $y \rightarrow D^4$, a partir da qual obtém-se o ramo dado na figura 6.4. Nesse ramo a terminação se encontra no mundo possível (i–4,j–6) como era de se esperar, pois a especificação pode ser falseada a partir do instante t = 5 como pode ser constatado pela comparação das saídas y e $\delta^4 u$ e visualizada num ramo do tableau nas figuras abaixo .

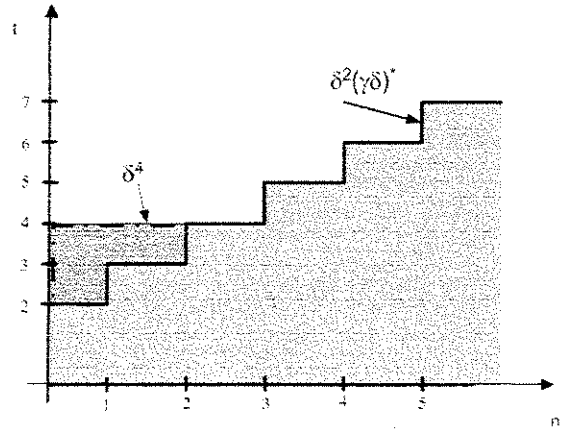


figura 6.3
Especificação $y \rightarrow D^4 u$ vista no plano

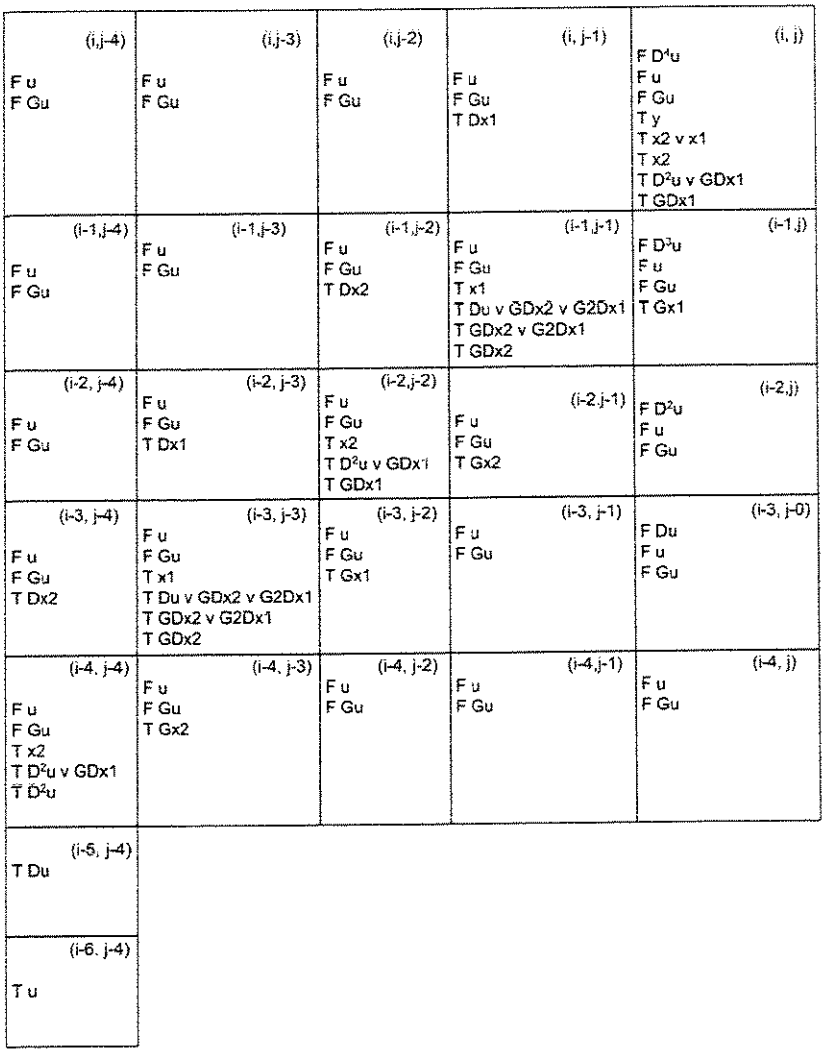


figura 6.4

Ramo aberto para a especificação $y \rightarrow D^4 u$ para o GET da fig. 6.1

Exemplo 2

Neste exemplo comparam-se duas saídas "y" e "z" do grafo da figura 6.5 abaixo onde as trajetórias das saídas $y = \delta^2(\gamma\delta)^*$ e $z = \delta(\gamma\delta^2)^*$ podem ser vistas na figura 6.6. Deseja-se saber se a trajetória z é mais "rápida" que a trajetória y, ou seja, se $z \leq y$. Traduzida na lógica NK, a especificação é $z \rightarrow y$.

O ramo aberto apresentado na figura 6.7 mostra que é possível falsear a especificação. Pode-se verificar que o tableau localizou a origem do plano n × t, (isto é, o ponto onde ocorre o impulso em u) no mundo possível (i–2, j–5). Esse ramo corresponde à situação em que o ponto de início do tableau (i–0, j–0) corresponde ao ponto (2, 5) do plano n×t. A especificação é falseada, pois neste ponto z>y.

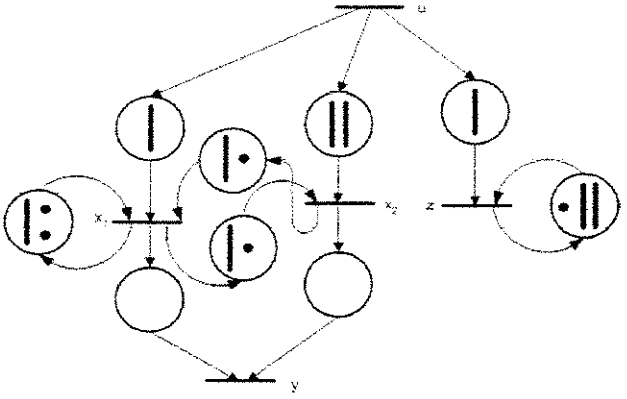


figura 6.5

Grafo para o exemplo 2

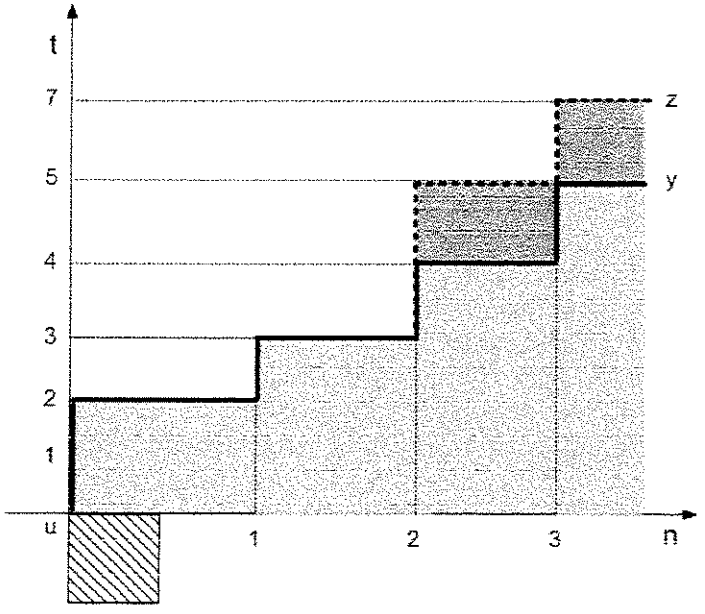


figura 6.6

Trajetórias das saídas y e z do grafo da fig. 6.5 no plano n×t

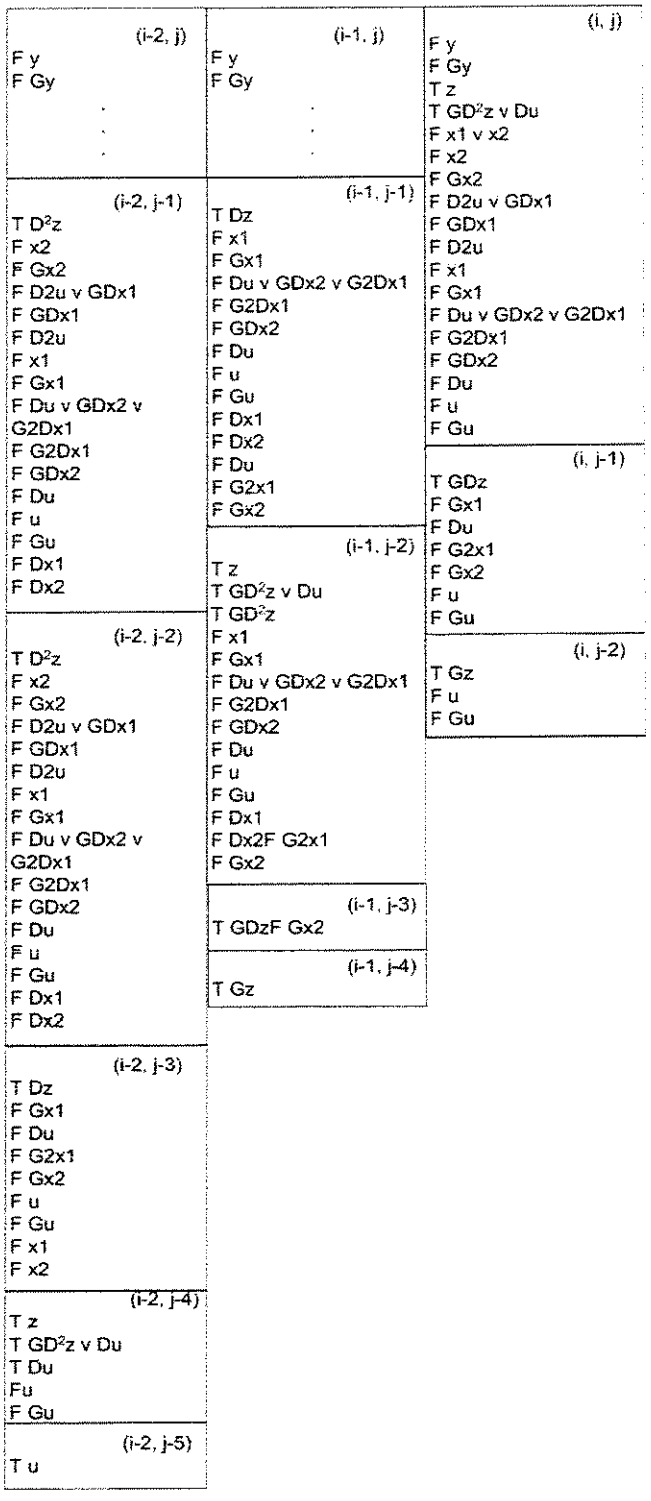


figura 6.7

Ramo aberto para a especificação z ⟶ y verificada para o GET da fig. 6.5

Exemplo 3

Neste exemplo, de maneira semelhante ao caso anterior, a especificação z→y compara as saídas y $= (\gamma^2\delta^2)^*$ e z $= \gamma\delta(\gamma^2\delta^2)^*$ do grafo dado pela figura abaixo.

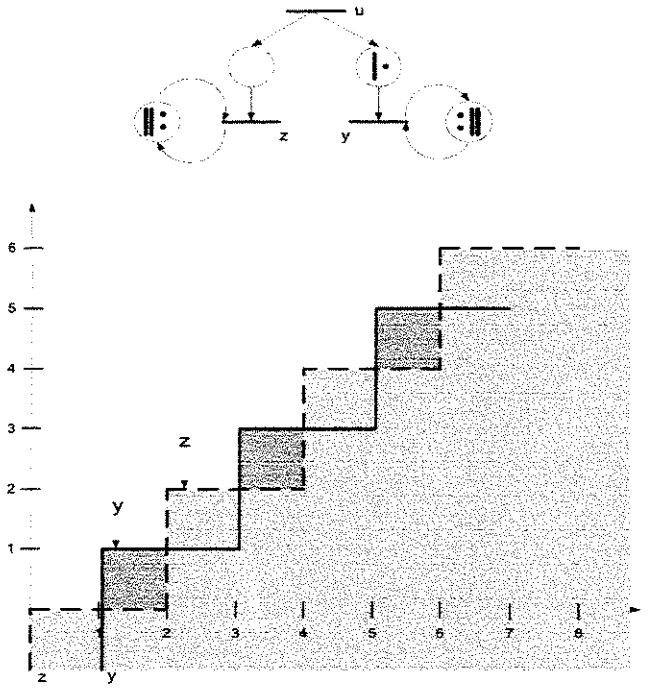


figura 6.8

GET para o exemplo 3 e as saídas z e y no plano n×t

A seguir temos os dois ramos abertos dados pelas figuras 6.9 e 6.10. terminados nos mundos (0, 0) e (i −2, j −2) respectivamente. Estes ramos em que o tableau localizou a terminação da variável de entrada "u" correspondem ás situações em que o tableau é iniciado respectivamente nos pontos (0, 0) e (2, 2) do plano n×t da figura 6.8. Nesses pontos a especificação é violada, confirmando o modelo correspondente aos ramos retornados pelo tableau.

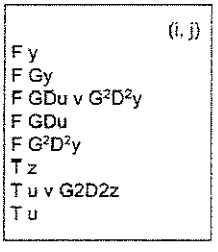


figura 6.9

Primeiro ramo aberto o GET do exemplo 3

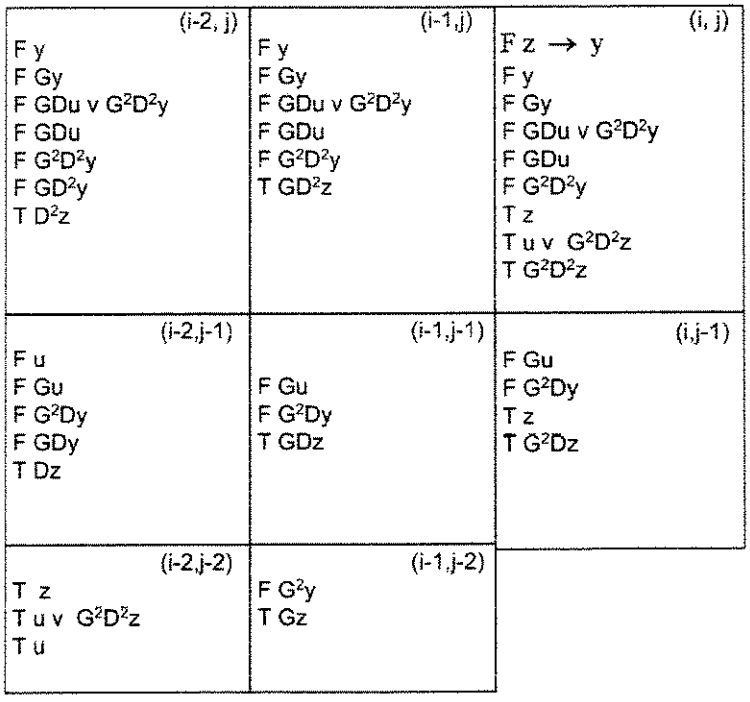


figura 6.10

Segundo ramo aberto o GET do exemplo 3

Exemplo 4

Um outro tipo de especificação pode ter por objetivo comparar uma dada saída y, de um grafo com a saída $z = y^*$ (o GET de saída z corresponde ao GET de saída y com realimentação pela transição k7). A especificação verificada é $z = y^* \leq y$ que equivale a determinar se $y = y^*$, pois em geral $y \leq y^*$ (ver definição 2.15). Portanto, se a especificação $z = y^* \leq y$ for atendida conclui-se que $y^* = y$. Este tipo de verificação é de grande importância em problemas de síntese de controladores (Cottenceau, 1999, Lüders, 2001).

Exemplo 5

Nesse caso a especificação é z→y isto é, deseja-se verificar se a saída z é "mais rápida" que a saída y. O GET proposto apresenta duas entradas u1 e u2 conforme a figura 6.14 onde z = $\delta^4(\gamma\delta^4)^*$u1 $\oplus$ $\delta^5(\gamma\delta^5)^*$u2 e y = $\delta^4(\gamma\delta^2)^*$u1 $\oplus$ $\delta^5(\gamma\delta^3)^*$u2 mostradas nas figuras 6.15 e 6.16. Pode-se considerar que a entrada localizada em situação de terminação corresponde a entrada impulsiva ($u_{terminada} = e$), e que a(s) outra(s) entrada(s) que não aparecem em situação de terminação no ramo correspondem a infinitas fichas desde um tempo $-\infty$ caracterizando a situação menos restritiva possível ($u_{não\ terminada} = \epsilon$) . Em outras palavras, nesse caso escreve-se a fórmula Tu não terminados em todos os mundos possíveis.

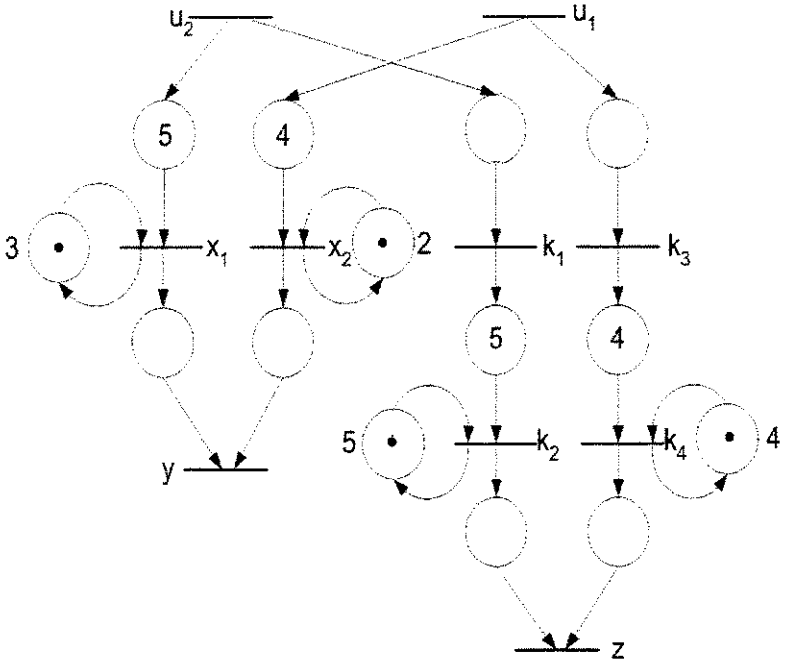


figura 6.14
GET com duas entradas u_1 e u_2

As trajetórias comparando z e y para ambas as entradas u_1 e u_2 podem ser vistas abaixo correspondendo aos índices dos mundos de terminação dos ramos abertos das figuras 6.17 a 6.20 encontrados pelo tableau aos pontos (1,8), (2, 12), para a variável u1 e (1, 10), (2, 15) para a variável u2, do plano n×t em que ocorre a violação da especificação verificada pelo tableau.

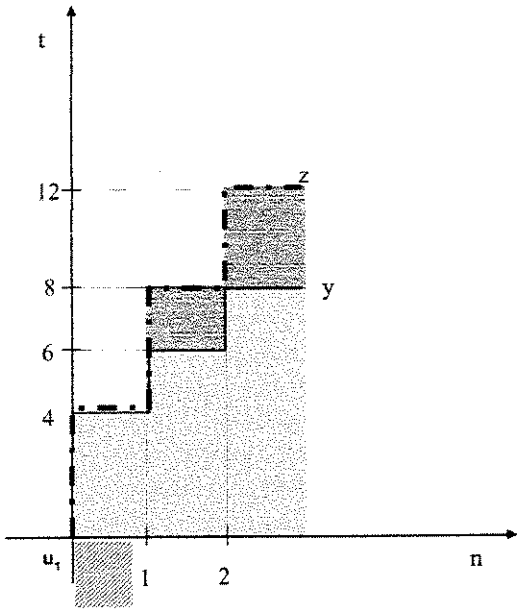


figura 6.15

Trajetória do GET da figura 6.14 em função da entrada u_1

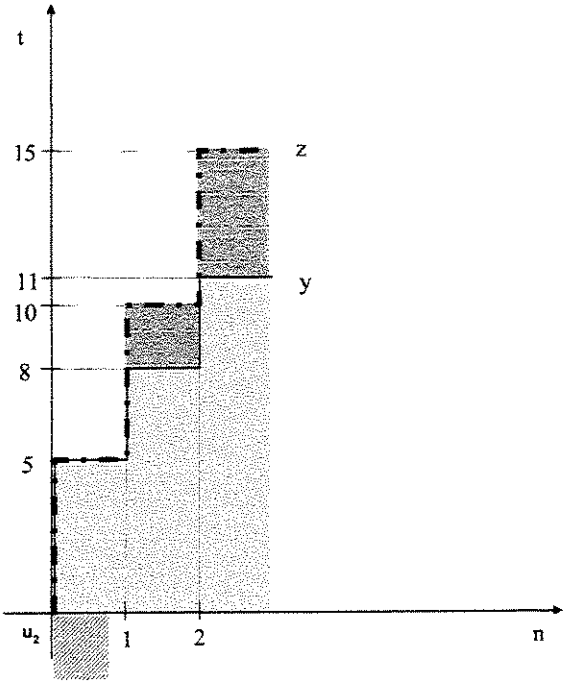


figura 6.16

Trajetória do GET da figura 5.14 em função da entrada u_2

(i-4,j)	(i-3, j)	(i-2,j)	(i-1,j)	F $z \rightarrow y$ (i,j) T z F y T $k_2 \vee k_4$ T k_2 T $GD^5k_2 \vee D^5k_1$ T GD^5k_2 F $x_1 \vee x_2$ F x_1 F x_2 F GD^5x_1 F GD^5x_2
(i-4,j-1)	(i-3,j-1)	(i-2,j-1)	(i-1,j-1)	(i,j-1)
(i-4,j-2)	(i-3,j-2)	(i-2,j-2)	(i-1,j-2)	(i,j-2)
(i-4,j-3)	(i-3,j-3)	(i-2,j-3)	F x_1 (i-1,j-3) F $GD^5x_1 \vee D^5u_2$ F GD^5x_1 F D^5u_2	(i,j-3)
(i-4,j-4)	(i-3,j-4)	(i-2,j-4)	(i-1,j-4)	(i,j-4)
(i-4,j-5)	(i-3,j-5)	(i-2,j-5)	T k_2 (i-1,j-5) T $GD^5k_2 \vee D^5k_1$ T D^5k_1	(i,j-5)
(i-4,j-6)	(i-3,j-6)	F x_1 (i-2,j-6) F $GD^5x_1 \vee D^5u_2$ F GD^5x_1 F D^5u_2	(i-1,j-6)	(i,j-6)
(i-4,j-7)	(i-3,j-7)	(i-2,j-7)	(i-1,j-7)	(i,j-7)
(i-4,j-8)	(i-3,j-8)	(i-2,j-8)	F u_2 (i-1,j-8)	(i,j-8)
(i-4,j-9)	(i-3,j-9)	(i-2,j-9)	(i-1,j-9)	(i,j-9)
(i-4,j-10)	(i-3,j-10)	(i-2,j-10)	T k_1 (i-1,j-10) T u_2	(i,j-10)

figura 6.17

Ramo terminado em (i–1, j–10) para a variável de entrada u_2

(i-4,j)	(i-3, j)	(i-2,j)	(i-1,j)	F $z \rightarrow y$ (i,j) T z F y T $k_2 \vee k_4$ T k_4 T $GD^4k_4 \vee D^4k_3$ T GD^4k_4 F $x_1 \vee x_2$ F x_1 F x_2 F GD^2x_1 F GD^2x_2 F $G^2D^2x_2$
(i-4,j-1)	(i-3,j-1)	(i-2,j-1)	(i-1,j-1)	(i,j-1)
(i-4,j-2)	(i-3,j-2)	(i-2,j-2)	(i-1,j-2) F x_2 F $GD^2x_2 \vee D^4u_1$ F GD^2x_2 F D^4u_1	(i,j-2)
(i-4,j-3)	(i-3,j-3)	(i-2,j-3)	(i-1,j-3) F x_1 F $GD^2x_1 \vee D^5u_2$ F GD^2x_1 F D^5u_2	(i,j-3)
(i-4,j-4)	(i-3,j-4)	(i-2,j-4) F x_2 F $GD^2x_2 \vee D^4u_1$ F GD^2x_2 F D^4u_1	(i-1,j-4) T k_4 T $GD^4k_4 \vee D^4k_3$ T GD^4k_4	(i,j-4)
(i-4,j-5)	(i-3,j-5)	(i-2,j-5)	(i-1,j-5)	(i,j-5)
(i-4,j-6)	(i-3,j-6) F x_2 F $GD^2x_2 \vee D^4u_1$ F GD^2x_2 F D^4u_1	(i-2,j-6)	(i-1,j-6)	(i,j-6)
(i-4,j-7)	(i-3,j-7)	(i-2,j-7)	(i-1,j-7)	(i,j-7)
(i-4,j-8)	(i-3,j-8)	T k_4 (i-2,j-8) T $GD^4k_4 \vee D^4k_3$ T D^4k_3	(i-1,j-8)	(i,j-8)
		•		
		•		
(i-4,j-12)	(i-3,j-12)	T k_3 (i-2,j-12) T u_1	(i-1,j-12)	(i,j-12)

figura 6.18

Ramo terminado em (i–2, j–12) para a variável de entrada u_1

(i-4,j)	(i-3, j)	(i-2,j)	(i-1,j)	F $z \rightarrow y$ (i,j) T z F y T $k_2 \vee k_4$ T k_4 T $GD^4k_4 \vee D^4k_5$ T GD^4k_4 F $x_1 \vee x_2$ F x_1 F x_2 F GD^3x_1 F GD^2x_2
(i-4,j-1)	(i-3,j-1)	(i-2,j-1)	(i-1,j-1)	(i,j-1)
(i-4,j-2)	(i-3,j-2)	(i-2,j-2)	(i-1,j-2) F x_2 F $GD^2x_2 \vee D^4u_1$ F GD^2x_2 F D^4u_1	(i,j-2)
(i-4,j-3)	(i-3,j-3)	(i-2,j-3)	(i-1,j-3) F x_1 F $GD^3x_1 \vee D^5u_2$ F GD^3x_1 F D^5u_2	(i,j-3)
(i-4,j-4)	(i-3,j-4)	(i-2,j-4) F x_2 F $GD^2x_2 \vee D^4u_1$ F GD^2x_2 F D^4u_1	(i-1,j-4) T k_4 T $GD^4k_4 \vee D^4k_5$ T D^4k_5	(i,j-4)
(i-4,j-5)	(i-3,j-5)	(i-2,j-5)	(i-1,j-5)	(i,j-5)
(i-4,j-6)	(i-3,j-6)	(i-2,j-6)	(i-1,j-6)	(i,j-6)
(i-4,j-7)	(i-3,j-7)	(i-2,j-7)	(i-1,j-7)	(i,j-7)
(i-4,j-8)	(i-3,j-8)	(i-2,j-8)	(i-1,j-8) T k_5 T u_1	(i,j-8)

figura 6.19

Ramo terminado em (i–1, j–8) para a variável de entrada u_1

(i-4,j)	(i-3, j)	(i-2,j)	(i-1,j)	F $z \rightarrow y$ (i,j) T z F y T $k_2 \vee k_4$ T k_2 T $GD^5k_2 \vee D^5k_1$ T GD^5k_2 F $x_1 \vee x_2$ F x_1 F x_2 F GD^5x_1 F GD^5x_2
(i-4,j-1)	(i-3,j-1)	(i-2,j-1)	(i-1,j-1)	(i,j-1)
(i-4,j-2)	(i-3,j-2)	(i-2,j-2)	(i-1,j-2)	(i,j-2)
(i-4,j-3)	(i-3,j-3)	(i-2,j-3)	(i-1,j-3) F x_1 F $GD^5x_1 \vee D^5u_2$ F GD^5x_1 F D^5u_2	(i,j-3)
(i-4,j-4)	(i-3,j-4)	(i-2,j-4)	(i-1,j-4)	(i,j-4)
(i-4,j-5)	(i-3,j-5)	(i-2,j-5)	(i-1,j-5) T k_2 T $GD^5k_2 \vee D^5k_1$ T GD^5k_2	(i,j-5)
(i-4,j-6)	(i-3,j-6)	(i-2,j-6) F x_1 F $GD^5x_1 \vee D^5u_2$ F GD^5x_1 F D^5u_2	(i-1,j-6)	(i,j-6)
		•		
(i-4,j-9)	(i-3,j-9) F x_1 F $GD^5x_1 \vee D^5u_2$ F GD^5x_1 F D^5u_2	(i-2,j-9)	(i-1,j-9)	(i,j-9)
(i-4,j-10)	(i-3,j-10)	(i-2,j-10) T k_2 T $GD^5k_2 \vee D^5k_1$ T D^5k_1	(i-1,j-10)	(i,j-10)
(i-4,j-11)	(i-3,j-11)	(i-2,j-11) F u_2	(i-1,j-11)	(i,j-11)
		•		
(i-4,j-15)	(i-3,j-15)	(i-2,j-15) T u_2	(i-1,j-15)	(i,j-15)

figura 6.20

Ramo terminado em (i–2, j–15) para a variável de entrada u_2

6.3 Conclusão

Neste capítulo foram apresentados diversos exemplos que ilustram a aplicação do tableau analítico para a verificação de especificações em GET's utilizando-se os resultados obtidos no capítulo 4 como critério de escolha da solução apresentada pelo ramo aberto do tableau, pois provou-se que o aparecimento de um ramo terminado corresponde a se encontrar uma solução mínima, causal e impulsiva. Para o caso de um GET possuir mais de uma entrada e o tableau retornar um ramo terminado em uma de suas variáveis de entrada indica que para aquela solução a variável de entrada corresponde ao impulso *e*, enquanto as demais variáveis de entrada que não aparecem no ramo podem ser entendidas como correspondendo a ϵ conforme explicado no exemplo 6 mostrando que esta característica do GET não impede a utilização do tableau. No apêndice 3 serão apresentados exemplos de verificação de especificações e as respectivas entradas de dados e saídas da implementação computacional descrita no capítulo 5.

Conclusão

Sistemas dinâmicos a eventos discretos se mostram relacionados aos mais diversos ambientes tecnológicos atuais, como linhas de montagem, redes de comunicação e computadores, tráfego de veículos etc. Entre as abordagens para a modelagem de SED's tem-se as redes de Petri, um formalismo conhecido para se descrever Sistemas a Eventos Discretos, mais especificamente, os GET's são uma classe de redes de Petri capazes de modelar sistemas discretos onde não há compartilhamento ou concorrência pelos recursos do sistema modelado. Diretamente relacionada aos GET's, a álgebra de dióides permite aplicar um tratamento algébrico à sua modelagem, permitindo descrever através de equações lineares numa álgebra não usual (idempotênte) a dinâmica de um GET. Um dos pontos de maior interesse na modelagem de SED's é o da síntese de controladores, que permitem aproximar a saída de um dado GET aproximando-a de uma dada especificação. O tratamento algébrico, quando aplicado a síntese de controladores sujeita-se a diversas restrições da algébra de dióides, impossibilitando a obtenção de controladores para GET's em que essas restrições não são satisfeitas, tornando a busca de abordagens alternativas para a modelagem de GET's de grande interesse. Uma das estratégias para se modelar GET's é através da lógica. A lógica proposicional modal NK foi proposta por Magossi (1998) para a modelagem de GET's utilizando-se do tableau analítico como ferramenta de verificação de especificações em GET's, tranformando uma verificação de especificações na álgebra de dióides em uma verificação de conseqüência lógica, via tableau, na lógica NK. O algoritmo proposto originalmente para o tableau analítico permitia ao tableau retornar soluções não-mínimas, não-causais e não-impulsivas, sendo essas soluções indesejáveis pois, as características da solução que se busca são a da minimalidade, causalidade e impulsividade.

O presente trabalho apresentou à lógica NK as seguintes contribuições: Um método de decisão para o tableau semântico capacitando-o a descartar ramos abertos associados à soluções não-mínimas, não-causais e não-impulsivas. Esse método está baseado nos teoremas apresentados no capítulo 4 que associam ao ramo dito terminado as características desejadas de minimalidade, causalidade e também se prova que se um dado ramo terminado não-impulsivo foi encontrado pelo tableau então um ramo terminado impulsivamente existe. Um algoritmo para o tableau é, então, proposto ao fim do capítulo 4 e sua implementação em uma linguagem de alto nível, no caso a linguagem de programação JAVA™, é apresentado. No capítulo 6 são vistos exemplos de verificação de especificações em GET's utilizando-se o algoritmo proposto enquanto os arquivos de entrada e as saídas da implementação computacional são vistas no anexo C.

Referências Bibliograficas.

Baccelli, F.L., G. Cohen, G. J. Olsder, J. P. Quadrat (1992). *Synchronization and Linearity - An Algebra for Discrete Event Systems.* John Wiley and Sons, New York.

Cassandras. C. G. (1993). *Discrete Event Systems: Modeling and Performance Analysis.* Aksen Associates Incorporated Publishers, Boston.

Cohen, G. ,P. Moller, J. P. Quadrat, M. Viot (1989). *Algebraic tools for performance evaluation of discrete systems.* Proceedings of IEEE, Vol. 77, no, 1, pp. 39-58, january.

Costa, M. M. C. (1992). *Introdução à Lógica Modal Aplicada à Computação.* VIII Escola de Computação. Instituto de Informática da UFRGS. Porto Alegre.

Cottenceau, B. (1999). *Contribuition à la Command de Systèmes à Evénements Discrets: Synthèse de Correcteurs pour les Graphes d'Evénements Temporisés dans les Dioïdes.* Thèse. Université d'Angers.

Dalen, D., Van (1980). *Logic and Structure.* Springer-Verlag, Berlin-Heidelberg.

Fitting, M. (1990). *First-Order Logic and Automated Theorem Proving.* Springer-Verlag, Berlin-Heildelberg.

Fitting, M. (1983). *Proof Methods for Modal and Intuitionistic Logics.* D. Reidel Publishing Conpany, Dordrecht.

Gaubert, S. (1992). *Théorie des Systèmes Linéaires dans les Dioïdes.* Thèse. École des Mines de Paris.

Hughes, G.E. e M.J. Cresswell (1968). *An Introduction to Modal Logic.* Methuen and Co. Ltd, London.

Hughes, G.E. e M.J. Cresswell (1984). *A Companion to Modal Logic.* Methuen and Co. Ltd, London.

Lüders, R. (2001). Controle Multivariável de Sistemas a Eventos Discretos em Dióides. Tese. Universidade Estadual de Campinas.

Magossi, J.C. (1998). Análise e Síntese de Sistemas a Eventos Discretos via Lógica Modal. Tese Universidade Estadual de Campinas.

Magossi, J.C., Santos Mendes, R. (1998) *Modal Logic Based Algorithms for the Verification os Specification in Discrete Event Systems.* International Workshop on Discrete Event Systems - WODES' 98, Cagliari, Italy, 508-513.

Mendelson, E. (1987). *Introduction to Mathematical Logic.* Chapman & Hall, New York.

Murata, T. (1989). *Petri Nets: Properties, analysis and applications.* Proceedings os IEEE, Vol. 77, pp. 541-580.

Ostroff, J. S. (1989). *Temporal Logic for Real Timed Systems.* John Wiley & Sons Inc. toronto

Pessanha, C. P. e R. Santos-Mendes (2003). "Modal Logic and Dioids"- Anais do "First Multidisciplinary International Symposium on Positive Systems" - POSTA 2003 - Roma, ITÁLIA, 31 - 38, 28 a 30/ago/2003

Priest, Graham. (2001) *An Itroduction to Non-Classical Logic* . Cambridge University Press, Cambridge.

Ramadge, P., W.M. Wonham (1989). *The control of discrete event systems.* Proceedings of IEEE, Vol 77, no.1, pp. 81-88

Smullyan, R. M. (1968). *First-Order Logic.* Springer-Verlag, Berlin_Heidelberg.

Anexo A

Lógica Proposicional Clássica e Lógica NK via Tableaux Analíticos.

A.1- Introdução

Neste apêndice será feita uma revisão dos conceitos de tableau clássico (Smullyan, 1968 e Fitting 1990) e do tableau modal proposicional NK (Magossi, 1998).

A.2- Árvores

O conceito de árvore binária está relacionado diretamente ao método de prova tableau, visto ser este elaborado na forma de árvores binárias conforme será definido adiante.

A.2.1 Definição 1: Árvores Não Ordenadas

Uma árvore não ordenada $\mathfrak{I}$ entende-se um conjunto S de pontos aos quais se associam:

- Uma função N que atribui a cada ponto α um inteiro positivo $N(\alpha)$ chamado nível de x.
- Uma relação x R y ("x é um predecessor de y") definida em S e satisfazendo os seguintes pontos:

a) Existe um único ponto de nível 1 chamado de origem da árvore.

b) Cada ponto que não seja a origem possúi um único predecessor.

c) Para quaisquer pontos α e β pertencentes a S, se $\alpha R\beta$ então $N(\beta) = N(\alpha)+1$. Δ

A figura abaixo ilustra os ítens acima, com a origem da árvore binária sendo α.

Árvore $\mathfrak{I}$

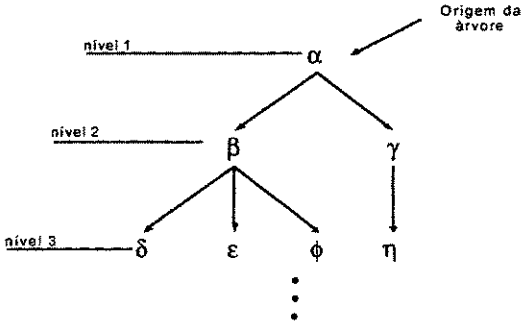


figura A 1

Representação de uma árvore

A.2.2 Definição 2: Árvores Ordenadas

Uma árvore ordenada $\mathfrak{I}$ é uma árvore não ordenada acrescida de uma função que associa à cada ponto de junção x da mesma uma seqüência de pontos sem repetição, sendo esses pontos todos os sucessores de x.△

Nota: Neste trabalho cada ponto de junção da árvore terá no máximo dois sucessores, caracterizando uma árvore **diádica**.

A.2.3 Definição 3: Árvore Finitamente Gerada.

Uma árvore $\mathfrak{I}$ é finitamente gerada se cada ponto tem um número finito de sucessores. △

A.2.4 Definição 4: Árvore Finita

Uma árvore $\mathfrak{I}$ é finita se possui um número finito de pontos, caso contrário será dita infinita.△

A.2.5 Definição 5: Ponto Final.

Um ponto de uma árvore é um ponto final se não possúi sucessores.△

A.3 Tableau Analítico.

O sistema de tableau para a lógica proposicional é devido a Hintikka (Hintikka, 1955) e foi uniformizado em sua notação por Smullyan (Smullyan, 1968). Esse sistema de prova se mostra mais versátil que o sistema axiomático pois, enquanto o sistema axiomático se vale de substituições para obter as provas desejadas, o sistema tableau se utiliza apenas das subfórmulas das fórmulas envolvidas na prova (veja figura A 2 abaixo).

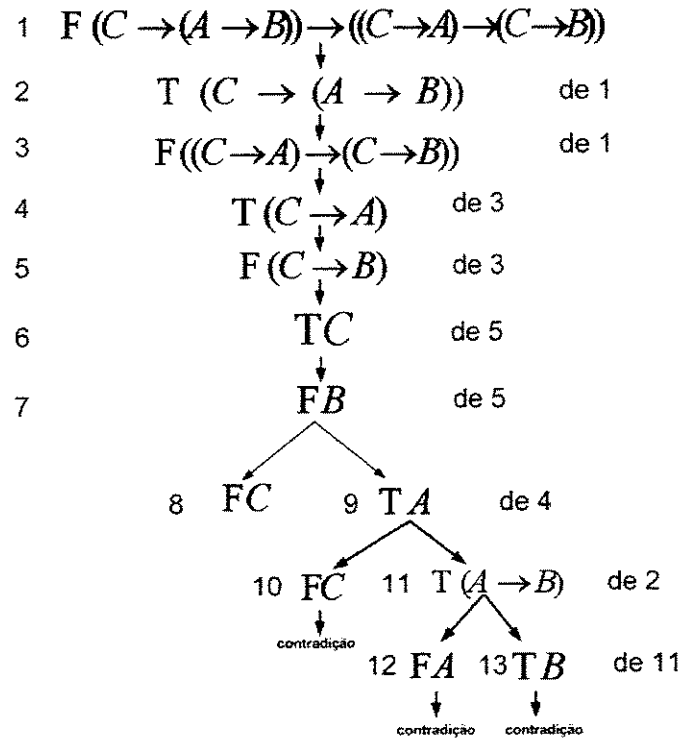


figura A.2

Ilustração de uma prova tableau

Na figura acima, pode-se notar que todas as 12 subfórmulas utilizadas na prova do teorema $(C \rightarrow (A \rightarrow B)) \rightarrow ((C \rightarrow A)(C \rightarrow B))$ estão contidas na própria fórmula que expressa o teorema.

A.4 Definição 6:

Uma fórmula assinalada é uma fórmula do tipo F X ou TX , sendo que X é uma fórmula. △

A.5 Regras de Extensão para o Tableau Analítico

As regras tableau podem ser divididas de acordo com o seu comportamento conjuntivo ou disjuntivo, ou seja, conjuntivas (do tipo α) são aquelas fórmulas que não causam uma bifurcação no ramo ou, em outras palavras, a abertura de um novo ramo. Já as fórmulas disjuntivas (do tipo β) são aquelas que, ao aplicadas, farão surgir no ramo uma bifurcação. Abaixo se encontram esquematizadas as regras α e β (a barra dupla || indica comportamento disjuntivo na aplicação da regra e, portanto, bifurcação de um ramo).

Regra $\alpha 1$:
T $(A \wedge B)$
T A
T B

Regra $\alpha 2$:
F $(A \vee B)$
F A
F B

Regra $\alpha 3$:
F$(A \rightarrow B)$
T A
F B

Regra $\alpha 4$:
F$\neg A$
T A

Regra $\beta 1$:
F $(A \wedge B)$
F A || F B

Regra $\beta 2$:
F $(A \vee B)$
F A || F B

Regra $\beta 3$:
T$(A \rightarrow B)$
F A || T B

Regra $\beta 4$:
T$\neg\alpha$
F α

A.6 Tableau Analítico: Descrição

O tableau analítico é um procedimento refutacional de prova elaborado na forma de árvores binárias sendo que estas contém um número finito de ramos e cada ramo é constituído de um dado número de nós onde cada nó é a ocorrência de uma fórmula assinalada. Para se verificar se uma dada fórmula é um teorema da lógica, ou não, insere-se essa mesma fórmula como nó inicial do tableau (como orígem do mesmo) assinalada com F (ou seja se X é a fórmula a ser provada, a orígem do tableau deve ser FX). De modo prático, trata-se de uma prova por absurdo iniciada com a negação daquilo que se deseja provar, ou seja, se se deseja provar que uma dada fórmula X é um teorema inicia-se o tableau afirmando FX e depois investiga-se, através das regras tableau, se a afirmação do nó inicial gera uma contradição. Como exemplo têm-se a seguir um tableau para a fórmula

$\neg(A \vee B) \rightarrow (\neg A \wedge \neg B)$.

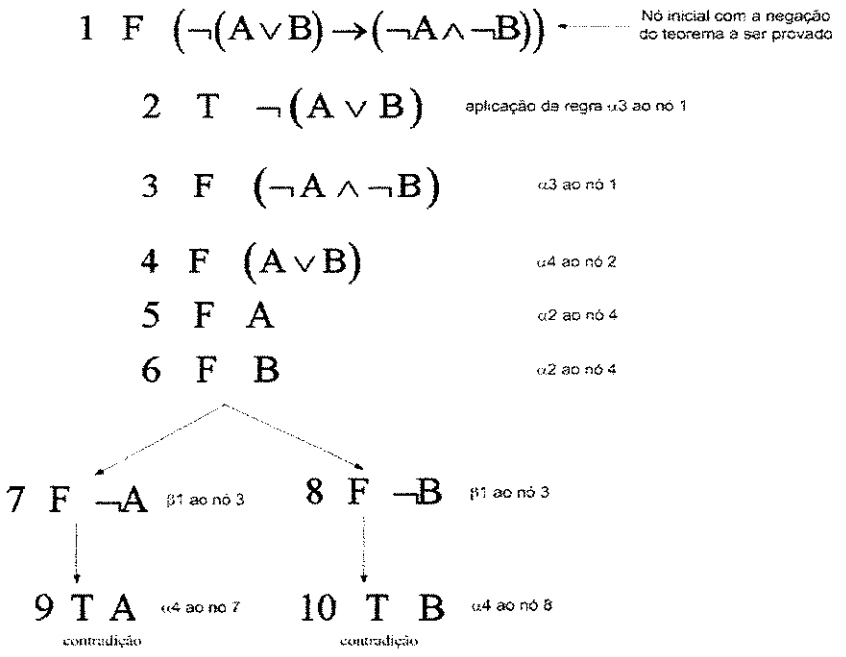


figura A.3

Prova tableau para a fórmula $\neg(A \vee B) \rightarrow (\neg A \wedge \neg B)$

Têm-se como nó inicial a negação da fórmula que se deseja provar. A partir daí aplica-se as regras tableau pertinentes. No caso do nó inicial F $(\neg(A \vee B)) \rightarrow ((\neg A \wedge \neg B))$, há pelas regras tableau apenas uma possibilidade advinda da afirmação de que uma implicação seja assinalada por "F" e que é a necessidade de ambos, antecedente e conseqüente sejam assinalados por T e F respectivamente conforme expressa a regra $\alpha 3$ inserindo-se daí os nós 2 e 3. O nó 3, que é F $\neg A \wedge \neg B$ e que, pela regra $\beta 1$ só pode acontecer se pelo menos um dos elementos da conjunção tiver assinalamento "F" gerando daí uma bifurcação. A análise continua desta mesma forma até que se encerrem as regras possíveis de serem aplicadas ou surja no ramo uma contradição encontrando-se no ramo, para alguma fórmula Tδ do ramo, Fδ, o que indica da mesma forma que numa prova por absurdo, que partiu-se de uma dada afirmação ao se negar a fórmula que se queria provar e chegou-se a uma situação absurda com uma mesma fórmula assinalada por T e F num mesmo ramo. Isto indica que, naquele ramo, não foi possível encontrar um modelo que confirme a afirmação inicial (a fórmula inicialmente negada). Se esta situação se repetir em **todos os ramos abertos pelo tableau** (diz-se que o tableau está **fechado**) têm-se que não foi possível encontrar um modelo que torne verdadeira a afirmação do nó inicial o que deixa apenas uma conclusão possível: A fórmula do nó inicial só pode ser verdadeira, haja visto que não foi

possível encontrar um modelo que a falseie. Abaixo é possível ver um exemplo de um tableau retornando uma valoração (ou modelo) para a fórmula testada:

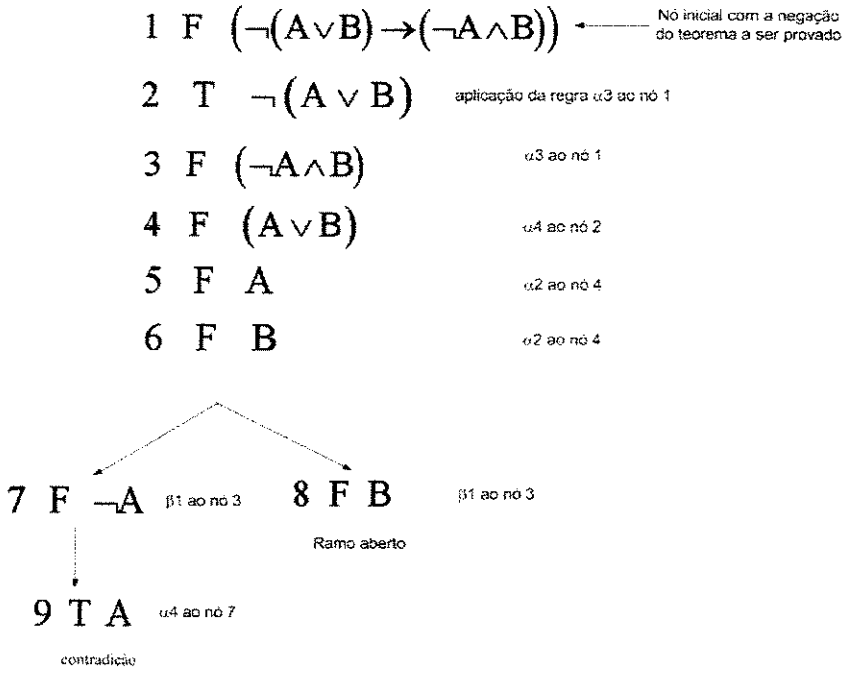


figura A.4

Um ramo tableau retornando um modelo para a fórmula $(\neg(A \vee B)) \rightarrow ((\neg A \wedge \neg B))$

Como se pode ver, no ramo da direita não se encontra uma contradição como no ramo da esquerda (onde se têm T A e F A). Como não há mais regras tableau que possam ser aplicadas no ramo da direita, têm-se que o ramo indica um modelo que torna correta a afirmação do nó inicial F($\neg(A \vee B)) \rightarrow ((\neg A \wedge B))$, o que pode ser verificado percorrendo-se o ramo e recolhendo-se os assinalamentos das variáveis A e B, que são FA e FB conforme se pode ver na tabela verdade abaixo:

	$(A \vee B)$	$\neg(A \vee B)$	$(\neg A \wedge B)$	$\neg(A \vee B) \rightarrow (\neg A \wedge B)$
$v(A) = f$ $v(B) = f$	f	t	f	f

Tabela A.1

Valores verdade retornados pelo ramo aberto pelo tableau para $\neg(A \vee B)) \rightarrow ((\neg A \wedge B)$

Acompanhando-se a tabela acima confirma-se a informação obtida via tableau, que a valoração FA e FB implica em F $\neg(A \vee B)) \rightarrow ((\neg A \wedge B)$ mostrando que a fórmula testada não é um teorema.

A.7 Conseqüência Lógica

Pode-se analisar casos de conseqüência lógica utilizando-se o tableau analítico. Se é o caso que se deseja verificar se D é conseqüência lógica de A, B e C deve-se inserir no tableau para análise a fórmula $(A \wedge B \wedge C) \rightarrow D$ e verificar se o tableau é fechado mostrando que não é possível ter-se FD e T$(A \wedge B \wedge C)$ que implica em TA, TB e TC. Em outras palavras, não é possível que os assinalamentos de A, B e C sejam T e o assinalamento de D seja F, logo deve-se ter, necessariamente, TD o que caracteriza a definição de conseqüência lógica: $A, B, C \vdash D$.

A.8 Definição de Tableau.

Um tableau analítico para uma dada fórmula Y é uma árvore ordenada do tipo diádica onde cada ponto marca a ocorrência de uma fórmula. Sendo construído de acordo com o procedimento abaixo:

1) Insira F Y na origem da árvore.

2) Se x é um ponto final o tableau pode ser extendido das seguintes maneiras:

2.1- Se alguma α-fórmula ocorrer no ramo cujo ponto final é x pode-se acrescentar α_1 e α_2 como os únicos sucessores de x.

2.2 - Se alguma β-fórmula ocorrer no ramo cujo ponto final é x pode-se acrescentar β_1 e β_2 como os únicos sucessores de x. $\triangle$

A.9 Prova de Correção do Tableau

Teorema 1

Se $\mathfrak{I}$ é um tableau satisfatível (que possua pelo menos um ramo com todas as fórmulas satisfatíveis). Se $\mathfrak{I}'$ é um tableau obtido de $\mathfrak{I}$ pela aplicação de uma regra de extensão de ramos do tipo α ou β, então $\mathfrak{I}'$ será satisfatível.

Demonstração:

Como o tableau é satisfatível, há pelo menos um ramo satisfatível entre todos os seus ramos. Se alguma regra de extensão de ramos for aplicada a algum outro ramo que não o ramo satisfatível do tableau, este continuaria satisfatível e, portanto, o tableau continuaria satisfatível. No caso da aplicação de alguma regra de extensão de ramos ao ramo satisfatível têm-se:

a) A regra de extensão de ramos aplicada é a regra-α:

Se A é a fórmula que sofrerá a aplicação da regra-α e S é o conjunto de todas as fórmulas do ramo diferentes de A, então o ramo conterá S $\cup \{A\}$, logo $\Im'$ apresentará um dos três subcasos :

i) S $\cup \{A, A_1\}$ ou

ii) S $\cup \{A, A_1, A_2\}$ ou

iii) S $\cup \{A, A_2\}$.

Como A_1 e A_2 são verdadeiras sempre que A o for (devido ao comportamento conjuntivo de α), logo em qualquer um dos casos $\Im'$ é satisfatível.

b) A regra de extensão de ramos aplicada é a regra β:

Se B é a fórmula que sofrerá a aplicação da regra-β e S é o conjunto de todas as fórmulas do ramo diferentes de B, então o ramo conterá S $\cup \{B\}$, logo $\Im'$ apresentará um dos três subcasos :

i) S $\cup \{B, B_1\}$ ou

ii) S $\cup \{B, B_2\}$.

Como B_1 ou B_2 são verdadeiras sempre que B o for (devido ao comportamento disjuntivo de B), o tableau terá um ramosatisfatível contendo S $\cup \{B, B_1\}$ ou S $\cup \{B, B_2\}$, logo em qualquer um dos casos $\Im'$ é satisfatível.▲

Teorema 2

Se uma fórmula x tem uma prova tableau entaõ x é uma tautologia.

Demonstração:

Suponha, por absurdo que x possua uma prova tableau e não seja uma tautologia.

Partindo da hipótese, se x tem uma prova tableau então o tableau para x é fechado mas, se x não é tautologia pode-se encontrar um modelo onde Fx. Como o tableau começa com Fx tem-se que o conjunto {Fx} é um conjunto satisfatível pois existe uma valoração tal que v(F x) = t. Pelo teorema anterior, qualquer aplicação das regras de extensão de ramos ao tableau será satisfatível indicando que o tableau iniciado com F x terá um ramo satisfatível, contrariando a hipótese inicial de que x teria uma prova tableau (todos os ramos do tableau estariam fechados). Logo, por absurdo, se x possui uma prova tableau então x é uma tautologia. ▲

A.10 Completude do Tableau

O objetivo será mostrar que se uma prova existe, então o tableau pode encontrá-la.

A.10.1 Definição 7
Ramo Completo

Um ramo de um tableau é dito completo se, para toda α-fórmula presente no ramo α_1 e α_2 também estão presentes no ramo e se , para toda β-fórmula presente no ramo β_1 ou β_2 também estão presentes no ramo. Δ

A.10.2 Definição 8
Ramo Completado

Um tableau $\Im$ é dito completado se cada ramo pertencente a $\Im$ é um ramo fechado ou completo. Δ

A.10.3 Definição 9
Conjunto de Hintikka

Um conjunto de fórmulas proposicionais K é um conjunto de Hintikka se :

1) para toda variável proposicional x não se dá que $x \in K$ e $\neg x \in K$.
2) Se uma fórmula do tipo α pertence a K então: $\alpha_1 \in K$ e $\alpha_2 \in K$.
3) Se uma fórmula do tipo β pertence a K então: $\beta_1 \in K$ ou $\beta_2 \in K$. Δ

A.11 Lema de Hintikka

Se H é um conjunto de Hintikka então H é satisfatível.

Demonstração:

Assume-se inicialmente o seguinte para uma variável p qualquer:

1) Se T p $\in$ H então v(p) = t.
2) Se F p $\in$ H então v(p) = f.
3) Se T p $\notin$ H e F p $\notin$ H então v(p) = t.

Prova por indução:

Base indutiva:

As fórmulas de grau 0 são verdadeiras por construção.

Hipótese indutiva:

Supondo-se que as fórmulas de grau k < n sejam verdadeiras prova-se que as de grau n também serão verdadeiras.

Caso 1: Fórmulas do tipo α. Se x é do tipo α, então α_1 e α_2 também pertencem a H. Como α_1 e α_2 são de grau menor que x, pela hipótese de indução, são ambas verdadeiras logo, x é verdadeira.

Caso 2: Fórmulas do tipo β. Se x é do tipo β, então β_1 e β_2 também pertencem a H. Como β_1 ou β_2 são de grau menor que x, e como ao menos uma delas pertence a H, pela hipótese de indução, ao menos uma delas é verdadeira, logo, x é verdadeira.

Portanto têm-se que um conjunto de Hintikka é satisfatível.▲

Teorema 3

Qualquer ramo completo aberto de qualquer tableau é satisfatível.

Demonstração:

Se um ramo $\Im$ é aberto então não existe uma fórmula $x \in \Im$ onde $x \in \Im$ e $\neg x \in \Im$ e se α é uma fórmula do tipo $-\alpha$ então $\alpha_1 \in \Im$ e $\alpha_2 \in \Im$ e se β é uma fórmula do tipo β então $\beta_1 \in \Im$ ou $\beta_2 \in \Im$. Logo, tem-se que um ramo aberto e completo é um conjunto de Hintikka, e portanto, pelo lema de Hintikka é satisfatível.▲

A.12 Completude
Teorema 4

Se x é uma tautologia, então tem uma prova tableau.

Demonstração:

Por contraposição.

Suponha que a prova tableau não exista. Então existe um ramo aberto completado e, portanto, satisfatível. O que implica que v(F x) = t ou v(x) = f, logo x não é uma tautologia o que contraria hipótese inicial.▲

Tableau Proposicional Modal Prefixado.

Diferentemente do tableau clássico, que se restringe a apenas um único mundo possível, o tableau para a lógica modal nk deve tratar de vários mundos possíveis acessíveis através de regras de acessibilidade (um mundo pode ser R-acessível ou S-acessível em relação a outro mundo possível). Portanto, será adotado o procedimento de criar nomes para cada mundo possível (ao contrário de um tableau para cada mundo possível) manipuláveis pelas regras de acessibilidade .

Definição 10
Prefixo.

Um prefixo é um par (i,j) tal que i, j $\in \mathbb{Z}$. Δ

Definição 11
Prefixo Usado.

Um prefixo $\sigma = (i, j)$ pode ser chamado de usado num ramo do tableau NK (para fórmulas prefixadas) se e somente se σ já acorreu anteriormente no ramo. Δ

Observação:

Conforme foi dito acima um prefixo representa um nome para um mundo possível têm-se que uma fórmula σX é verdadeira numa dada interpretação, se X for verdadeira no mundo cujo nome é σ.

Definição 12
R-acessibilidade e S - acessibilidade.

Um prefixo $\sigma' = (r, s)$ é dito R – acessível a partir de outro prefixo $\sigma = (i, j)$ se e somente se $r = i-1$ e $s = j$. De maneira análoga $\sigma' = (r, s)$ é dito S – acessível a partir de outro prefixo $\sigma = (i, j)$ se somente se $r = i$ e $s = j-1$. Δ

Como cada prefixo pertence a $\mathbb{Z}^2$, têm-se que a partir de qualquer $\sigma \in \mathbb{Z}^2$ sempre existirá um prefixo σ', S-acessível ou um prefixo σ', R-acessível

Se considerarmos (i,j) e (r,s) pertencentes a $\mathbb{Z}^2$ e ao conjunto W definido anteriormente, então, temos para $\Re$ e S:

- (i, j) **R** (r,s) se e somente se $r = i-1$ e $j = s$.
- (i, j) **S** (r,s) se e somente se $r = i$ e $j = s-1$.

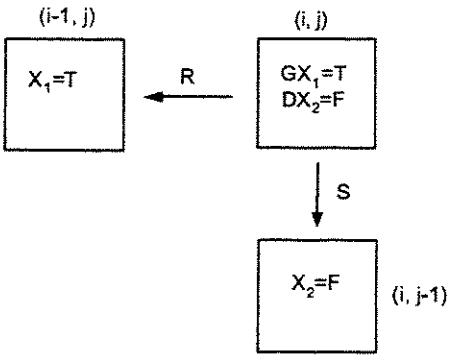


figura A.5

Ilustração da acessibilidade entre os mundos possíveis através das relações R e S

Como cada w está associado a um ponto do plano cartesiano, indicado por (i, j), os itens relativos à V(GX, w) = t e V(DX, w) = t acima podem ser reescritos como (veja a figura abaixo):

(i, j) $\models$ GX sse ($\exists$(r, s) $\in$ W) ((i, j)**R**(r, s) e (r, s)$\models$ X)

(i, j) $\models$ DX sse ($\exists$(r, s) $\in$ W) ((i, j)**S**(r, s) e (r, s) $\models$ X)

Nota: (X,w) = t nota-se como w $\models$ X (w força X).

As regras para manipulação dos mundos serão chamadas de:

Regra ν: Aplicável ao operador G.

Regra π: Aplicável ao operador D.

k	ν	(i,j)		k	π	(i,j)
k+n	ν_0	(i-1,j)		k+n	π_0	(i,j-1)

figura A.6

Aplicação das regras ν e π em função das linhas do tableau

Em termos dos operadores modais G e D tem-se:

ν - fórmula	
ν	ν_0
T GX	TX
F GX	FX

π - fórmula	
π	π_0
T DX	TX
F DX	FX

Tabela A.2

Regras ν e π em função dos valores verdade

Definição 13
Correção

Sejam um conjunto S de fórmulas prefixadas e $\langle W,R,S,v\rangle$ um modelo para a lógica modal nk. Uma **interpretação** de S num modelo é dada por uma aplicação I que associa a cada σ pertencente ao conjunto de prefixos de S um mundo possível pertencente a W, tal que:

- Se σ' é R-acessível a partir de σ então I(σ) R I(σ').
- Se σ' é S-acessível a partir de σ então I(σ) S I(σ').
- Onde I(σ) é um mundo de W enumerado pelo prefixo σ. △

Definição 14

Diz-se que um conjunto S de fórmulas prefixadas é satisfatível sob uma interpretação I se, para cada σZ∈S for verdade que I(σ) ⊨ Z, logo, S é satisfatível se S é satisfatível sob alguma interpretação. △

Definição 15
Ramo satisfatível

Um ramo de um tableau é satisfatível se o conjunto de fórmulas prefixadas que ele contém for satisfatível. △

Definição 16

Um tableau é satisfatível se, pelo menos um ramo for satisfatível. △

Lema 1

Seja T um tableau prefixado que é satisfatível. Seja T' um tableau que resulta de uma simples regra tableau aplicada a T. Então T' é satisfatível

Demonstração:

Se T é um tableau satisfatível devido a um ramo $\mathfrak{I}$ satisfatível, se uma regra de extensão de ramos for aplicada a algum ramo $\mathfrak{I}' \neq \mathfrak{I}$ tem-se que $\mathfrak{I}'$ ainda é satisfatível, pois ainda possui um ramo satisfatível.

Análise dos casos de aplicação de uma regra de extensão ao ramo $\mathfrak{I}$:

Caso 1: Regra α.

Se a partir de $\mathfrak{I}$ a regra α é aplicada e S é o conjunto de todas as fórmulas de $\mathfrak{I}$ diferentes de α o ramo $\mathfrak{I}$ contém o cunjunto $S \cup \{\alpha\}$. Daí $\mathfrak{I}'$ se adequará a um dos três subcasos:

1) $S \cup \{\alpha, \alpha_1\}$ ou

2) $S \cup \{\alpha, \alpha_1, \alpha_2\}$ ou

3) $S \cup \{\alpha, \alpha_2\}$.

Mas α_1 e α_2 são verdadeiras sempre que α o for (comportamento conjuntivo). Logo, em todos os 3 casos $\mathfrak{I}'$ será satisfatível.

Caso 2: Regra β.

Se a partir de $\mathfrak{I}$ a regra β é aplicada e S é o conjunto de todas as fórmulas de $\mathfrak{I}$ diferentes de β o ramo $\mathfrak{I}$ contém o cunjunto $S \cup \{\beta\}$. Daí $\mathfrak{I}'$ se adequará a um dos três subcasos:

1) $S \cup \{\beta, \beta_1\}$ ou

2) $S \cup \{\beta, \beta_2\}$.

Mas β_1 e β_2 são verdadeiras sempre que β o for (comportamento disjuntivo). Logo, em todos os 2 casos $\mathfrak{I}'$ será satisfatível.

Caso 3: Regra ν.

Suponha que a fórmula $\sigma\nu$ ocorre no ramo $\mathfrak{I}$ e que a regra ν é aplicada adicionando a fórmula $\sigma'\nu_0$ no fim do ramo $\mathfrak{I}$. Como o ramo $\mathfrak{I}$ é satisfatível então existe uma I-interpretação onde σ' é R-acessível a partir de σ portanto $I(\sigma)\ R\ I(\sigma')$. Mas $\sigma\nu \in \mathfrak{I}$ que, por sua vêz é satisfatível implicando que $I(\sigma) \models \nu$ e, pela definição de R-acessibilidade tem-se que $I(\sigma') \models \nu_0$ implicando finalmente que o ramo $\mathfrak{I}'$ é satisfatível.

Caso 4: Regra π.

Suponha que a fórmula $\sigma\pi$ ocorre no ramo $\mathfrak{I}$ e que a regra π é aplicada adicionando a fórmula

$\sigma' \pi_0$ no fim do ramo $\Im$. Como o ramo $\Im$ é satisfatível então existe uma I-interpretação onde σ' é R-acessível a partir de σ portanto I(σ) R I(σ'). Mas $\sigma\pi \in \Im$ que, por sua vêz é satisfatível implicando que I(σ) $\models \pi$ e, pela definição de R-acessibilidade tem-se que I(σ') $\models \pi_0$ implicando finalmente que o ramo $\Im'$ é satisfatível.

Logo $\Im'$ é satisfatível e portanto T' é satisfatível.▲

Correção Fraca
Teorema 5

Se uma fórmula X tem uma prova tableau prefixada, então x é válida em todos os modelos $\langle W, R, S, v \rangle$.

Demonstração:

Supõe-se, por absurdo, que X tem uma prova tableau, mas X não é valida. A primeira linha do tableau é F X (i, j) e como partiu-se de hipótese que x não é válida deve-se ter um mundo em um modelo $\langle W, R, S, v \rangle$ em que X não é forçada em um mundo w, logo o conjunto formado por {F X} é satisfatível, e o nó (1) F X (i, j) é satisfatível. Pelo lema anterior cada extensão de um tableau satisfatível é satisfatível. mas um tableau satisfatível não pode ser fechado o que contradiz a hipótese inicial que X possui uma prova tableau. Logo, se X possui uma prova tableau então X é válida em todos os modelos.▲

Definição 17
Completude

Um ramo θ de um tableau para fórmulas prefixadas é dito completo se e somente se as seguintes condições são satisfeitas:

- Se $\sigma\alpha \in \theta$ então $\sigma\alpha_1 \in \theta$ e $\sigma\alpha_2 \in \theta$.
- Se $\sigma\beta \in \theta$ então $\sigma\beta_1 \in \theta$ ou $\sigma\beta_2 \in \theta$.
- Se $\sigma\nu \in \theta$ então $\sigma\nu_0' \in \theta$ para $\sigma R \sigma'$.
- Se $\sigma\pi \in \theta$ então $\sigma\pi_0' \in \theta$ para $\sigma S \sigma'$.△

Definição 18
Tableau Completado

Um tableau prefixado é completado se e somente se cada ramo é completo ou fechado.

Um tableau completado é aquele no qual todas as regras que poderiam ser aplicadas foram aplicadas.△

Definição 19
Conjunto de Fórmulas Saturado

Um conjunto S de fórmulas prefixadas é saturado se:

- Não se tem σTA e σFA para qualquer fórmula atômica A e prefixa σ.
- Se $\sigma\alpha \in$ S então $\sigma\alpha_1 \in$ S e $\sigma\alpha_2 \in$ S.
- Se $\sigma\beta \in$ S então $\sigma\beta_1 \in$ S ou $\sigma\beta_2 \in$ S.
- Se $\sigma\nu \in$ S então $\sigma'\nu_0 \in$ S para $\sigma\, R\, \sigma'$.
- Se $\sigma\pi \in$ S então $\sigma'\pi_0 \in$ S para $\sigma\, S\, \sigma'$.△

Lema 2

Cada ramo aberto e completo de um tableau para fórmulas prefixadas é um conjunto saturado.

Demonstração:

Seja $\Im$ um ramo aberto e completo de um tableau para fórmulas prefixadas. Como o ramo é aberto não existe uma fórmula atômica A onde σTA e σFA pertencem ao ramo $\Im$ satisfazendo primeira condição de um conjunto saturado. Como o ramo é, por hipótese, completo a segunda e terceira condições são satisfeitas de modo direto. Como W percorre $\mathbb{Z}^2$ sabe-se que, para qualquer prefixo σ existe sempre um prefixo σ' onde $\sigma\, R\, \sigma'$ satisfazendo a quarta condição. De modo análogo a quinta condição também é satisfeita. Logo Cada ramo aberto e completo de um tableau para fórmulas prefixadas é um conjunto saturado. ▲

Teorema 6
Completude Fraca

Se X é válida em todos os modelos $\langle W, R, \mathrm{S}, \nu\rangle$ então cada tableau completado começado por σFA deve fechar em um sistema de fórmulas prefixadas.

Demonstração:

Suponha que X não tenha uma prova tableau (tem-se um ramo aberto). Pelo lema anterior esse ramo é um conjunto saturado e como cada conjunto saturado é satisfatível esse ramo aberto também é

satisfatível. Logo a origem também é satisfatível, o que implica que X não é válida o que contraria hipótese inicial. Logo se X é válida também possui uma prova tableau.▲

Teorema 7
Correção e Completude Fortes

Sejam S um conjunto de fórmulas não assinaladas e X uma fórmula não assinalada. Então $S \vdash_{dt} X$ se e somente se $S \models X$.

Demonstração:

- A parte $S \vdash_{dt} X \rightarrow S \models X$ (correção forte) é análoga ao teorema de correção fraca já feito anteriormente.

- A parte $S \models X \rightarrow S \vdash_{dt} X$ (completude forte)

Suponha, por absurdo que exista ao menos um ramo $\Im$ aberto no tableau (ou seja $S \nvdash_{dt} X$).

Estágio 0:

Começa-se o tableau por (1) F X (i, j) e completa-se o tableau.

Estágio n:

Adiciona-se ao fim de cada ramo $\Im$ aberto do tableau para cada prefixo σ_i onde $i \leqslant n$ as fórmulas prefixadas $\sigma_i TA_1, \sigma_i TA_2, \sigma_i TA_3, \sigma_i TA_4 \ldots \sigma_i TA_n$ até completar-se o tableau. Ao final do procedimento sistemático parqa n = 1,2,3...n o tableau vai estar aberto. Tem-se portanto um ramo aberto e que é ao mesmo tempo um conjunto saturado devido ao procedimento usado. Se o procedimento sistemático terminar o ramo que é aberto e saturado é também satisfatível (veja o lema 3.44 em Magossi, 1998). Por outro lado se o procedimento não terminar, pelo lema de Konig (Smullyan, 1968) esse tableau tem um ramo infinito e esse ramo infinito é aberto e saturado e portanto satisfatível. Mas como o conjunto de fórmulas prefixadas em $\Im$ contém (1) F X (i, j) e também contém σT Z para cada $Z \in S$ tem-se que F X é satisfatível logo X é falsa e posssui um modelo. Logo $S \nvDash X$. O que completa a prova.▲

Anexo B

Correspondência entre a Lógica NK e o Dióide *M*.

B.1 Introdução.

Neste anexo as provas dos teoremas que possibilitam a correspondência entre o dióide M e a lógica NK demonstrados originalmente em Magossi (1998) são apresentadas.

Teorema 1

Para a, b e $(a \oplus b) \in M_H$ então, a seguinte fórmula de NK é válida:

$I(a \oplus b) \leftrightarrow (I(a) \vee I(b))$.

Demonstração:

Seja $(i, j) \in W$ um mundo qualquer de uma I-interpretação; então:

$v((i, j), I(a \oplus b)) = t$

sse

$(a \oplus b) \geq \gamma^i \delta^j$ (da definição de I-interpretação)

sse

$a \geq \gamma^i \delta^j$ ou $b \geq \gamma^i \delta^j$ (propriedade do dióide M)

sse

$v((i, j), I(a)) = t$ ou $v((i, j), I(b)) = t$ (da definição de I-interpretação)

sse

$v((i, j), (I(a) \vee I(b))) = t$ (da semântica de NK)

Provou-se portanto que:

$v((i, j), I(a \oplus b)) = t$ se e somente se $v((i, j), (I(a) \vee I(b))) = t$

$\models_I I(a \oplus b) \leftrightarrow (I(a) \vee I(b))$ ▲

Teorema 2

Para a, $\gamma^n \delta^d a \in M_H$ e $n, d \in \mathbb{Z}^+$ então, é válida a seguinte fórmula:

$G^n D^d I(a) \leftrightarrow I(\gamma^n \delta^d a)$

Demonstração

Seja (i, j) ∈ W um mundo qualquer de uma I-interpretação; então:

$v((i, j), G^n D^d I(a)) = t$

sse

$v((i-n,j-d), I(a)) = t$ (da definição dos operadores G e D)

sse

$a \geq \gamma^{i-n}\ \delta^{j-d}$ (da definição de I-interpretação)

sse

$\gamma^n\ \delta^d a \geq \gamma^i\ \delta^j$ (multiplicando os dois lados da inequação por $\gamma^n\ \delta^d$)

sse

$v((i, j), I(\gamma^n\ \delta^d a)) = t$ (da definição de I-interpretação)

Portanto, para qualquer I-interpretação, para qualquer (i, j):

$v((i, j), G^n D^d I(a)) = t$ se e somente se $v((i, j), I(\gamma^n\ \delta^d a)) = t$

Da semântida de NK, conclui-se que:

$\models_I G^n D^d I(a) \leftrightarrow I(\gamma^n\ \delta^d a)$ ▲

Teorema 3

Para qualquer $a \in M_H$, as duas fórmulas seguintes são verdadeiras:

a) $GI(a) \rightarrow I(a)$

b) $I(a) \rightarrow DI(a)$

Demonstração

Prova de a)

Seja (i, j) ∈ W um mundo qualquer de uma I-interpretação; então:

$v((i, j), GI(a)) = t$

sse

$v((i-1, j), I(a)) = t$ (da definição do operador G)

sse

$a \geq \gamma^{i-1}\delta^{j}$ (multiplicando a inequação por γ^{-1})

Se $a \geq \gamma^{i-1}\delta^{j}$ então $a \geq \gamma^{i}\delta^{j}$ (propriedade do dióide M)

logo $a \geq \gamma^{i}\delta^{j}$

sse

$v((i, j), I(a)) = t$ (da definição de I-interpretação)

Portanto, para qualquer I-interpretação, para qualquer (i, j):

Se $v((i, j), GI(a)) = t$ então $v((i, j), I(a)) = t$

Conclui-se, portanto:

$\models_I GI(a) \rightarrow I(a)$ ▲

Prova de b)

Seja $(i, j) \in W$ um mundo qualquer de uma I-interpretação; então:

$v((i, j), I(a)) = t$ (da definição de I-interpretação)

sse

$a \geq \gamma^{i}\delta^{j}$

Se $a \geq \gamma^{i}\delta^{j}$ então $a \geq \gamma^{i}\delta^{j-1}$ (propriedade do dióide M)

logo $a \geq \gamma^{i}\delta^{j-1}$

sse

$v((i, j-1), I(a)) = t$ (da definição de I-interpretação)

sse

$v((i, j), DI(a)) = t$ (da definição do operador D)

Portanto, para qualquer I-interpretação, para qualquer (i, j):

Se $v((i, j), I(a)) = t$ então $v((i, j), DI(a)) = t$

Conclui-se, portanto:

$\models_I I(a) \rightarrow DI(a)$ ▲

Teorema 4

Para qualquer $a \in M_H$, $a \geq b$ se e somente se a seguinte fórmula da lógica NK for válida:

$I(b) \rightarrow I(a)$

Demonstração

Seja (i, j) ∈ W um mundo qualquer de uma I-interpretação; então:

$v((i, j), (I(b) \longrightarrow I(a))) = t$

sse

$v((i, j), I(b)) = f$ ou $v((i, j), I(a)) = t$ (da semântica de NK)

sse

não é o caso que $b \geq \gamma^i \delta^j$ ou $a \geq \gamma^i \delta^j$ (da definição de I-interpretação)

sse

$a \geq b$ (do lema 4.8 de Magossi, 1998)

O que completa a demonstração.▲

Teorema 5

Se $\mathcal{A}$ é uma valoração da lógica NK tal que as fórmulas $Gp \rightarrow p$ e $p \rightarrow Dp$ sejam satisfeitas para todas as variáveis proposicionais em todos os mundos sob a valoração $\mathcal{A}$, então a valoração $\mathcal{A}$ é uma I-interpretação.

Demonstração

Seja (i, j) ∈ W um mundo qualquer na valoração $\mathcal{A}$ e sejam r e s números inteiros tais que $r \geq i$ e $s \leq j$;

se $\mathcal{A}((i, j), p) = t$ então $\mathcal{A}((i+1, j), Gp) = t$ (da definição da semântica operador G)

se $\mathcal{A}((i+1, j), Gp) = t$ então $\mathcal{A}((i+1, j), p) = t$ (da hipótese do teorema)

se $\mathcal{A}((i+1, j), p) = t$ então $\mathcal{A}((i+2, j), Gp) = t$ (da definição da semântica do operador G)

⋮

se $\mathcal{A}((r, j), Gp) = t$ então $\mathcal{A}((r, j), p) = t$ (da hipótese do teorema)

⋮

se $\mathcal{A}((r, j), p) = t$ então $\mathcal{A}((r, j), Dp) = t$ (da hipótese do teorema)

se $\mathcal{A}((r, j), Dp) = t$ então $\mathcal{A}((r, j-1), p) = t$ (da definição da semântica do

operador D)

⋮

se $\mathcal{A}$ ((r, s+1), Dp) = t então $\mathcal{A}$ ((r, s),p) = t (da definição da semântica do operador G)

Portanto, se $\mathcal{A}$ ((i, j),p) = t então $\mathcal{A}$ ((r, s),p) = t.

Para mostrar que a valoração $\mathcal{A}$ é uma I-interpretação deve-se mostrar que se as equações Gp → p e p → Dp são satisfeitas para cada variável proposicional p então existe um elemento $a \in M$ que pode ser associado à variável proposicional p através de uma função p = I(a) de modo que $\mathcal{A}$ ((i, j), p) se e somente se $a \geq \gamma^i \delta^j$. Isto é feito construindo-se tal elemento. considere-se o conjunto dos pares ordenados de números inteiros $P = \{(i, j) \in \mathbb{Z}^2 \ / \ \mathcal{A}((i, j),p) = t\}$. O elemento desejado por: $a = \sum_{(i,j)\in p} \gamma^i \delta^j$. Mostra-se a seguir que $\mathcal{A}$ ((i, j),p) = t se e somente se $a \geq \gamma^i \delta^j$ o que prova que A é uma I-interpretação.

a) Se $\mathcal{A}$ ((i, j),p) = t então $a \geq \gamma^i \delta^j$ pela própria construção de a.

b) Se $a \geq \gamma^i \delta^j$ então $a = \sum_{(i,j)\in p} \gamma^i \delta^j$; portanto:

$\exists (i, j) \in P / \gamma^i \delta^j \geq \gamma^r \delta^s$.

Logo,

$\mathcal{A}$ ((i, j),p) = t e r ≥ i e s ≤ j.

portanto, pelo resultado do início da demonstração:

$\mathcal{A}$ ((r, s),p) = t, conclui-se portanto que:

$\mathcal{A}$ ((i, j),p) = t se e somente se $a \geq \gamma^i \delta^j$. O que prova que p = I(a) e que a valoração é uma I-interpretação.▲

Teorema 6

Sejam x_i, com i = 1,2,...,n elementos do dióide M, então o conjunto Θ de equações do tipo $\Theta = \left\{ x_i = \sum_{j=1}^{n} \gamma^{m_{ij}} \delta^{d_{ij}} x_j \ , \ i=1,2,3,..,n \right\}$ é satisfeito se e somente se o conjunto Ω de NK fórmulas notado por:

$$\Omega = \left\{ \begin{array}{c} pi \leftrightarrow \bigvee_{j=1}^{n} G^{m_{ij}} D^{d_{ij}} p_j \\ Gp_i \rightarrow p_i \\ p_i \rightarrow Dp_i \\ (i=1,2,3,...n) \end{array} \right\} \text{for verdadeiro em todos os mundos de alguma valoração.}$$

Demonstração

a) Se Θ é satisfeito, então as fórmulas de Ω são verdadeiras.

Seja a função H de ligação e sua correspondente I-interpretação. Fazendo-se a variável proposicional $p_i = I(x_i)$ e a variável proposicional $p_j = I(x_j)$ tem-se, dos teoremas 1, 2 e 3 que as fórmulas do conjunto Ω são verdadeiras para a I-interpretação, e portanto, verdadeiras para todo (i, j) $\in \mathbb{Z}^2$.

b) Se as fórmulas do conjunto Ω são verdadeiras para todo (i, j) $\in \mathbb{Z}^2$. então a função I é uma I-interpretação pelo teorema anterior. Portanto, pelos teoremas 1 e 2 o conjunto Θ de fórmulas satisfeito. ▲

Anexo C

Exemplos de Verificações de Especificações no Programa Tableau.

C.1 Introdução

Neste anexo serão mostrados exemplos de especificações verificadas com o auxílio da implementação computacional do algoritmo tableau proposto no capítulo 4. Inicialmente serão mostrados os resultados de verificações de especificações em alguns grafos com uma entrada 'u', posteriormente um exemplo de grafo com múltiplas entradas será mostrado para que se possa discutir como o tableau trata verificações de especificações em grafos com essa característica.

Exemplo 1

Inicialmente será utilizado o grafo com saída $y = \delta^2(\gamma\delta)^*$.

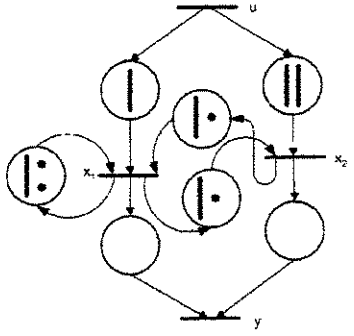


figura C.1
GET para o exemplo 1

a saída pode ser vista no plano abaixo:

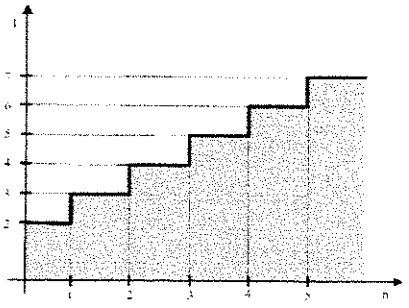


figura C.2

Trajetória do GET da figura C.1 no plano n×t

Ao verificarmos uma especificação do tipo y $\rightarrow D^4u$ obtemos a saída na forma do ramo mostrado abaixo, onde têm-se a terminação do ramo detectada no mundo possível (i-4,j-6) como era de se esperar, pois a especificação pode ser falseada a partir do instante t = 5 (inclusive) como pode ser constatado pela comparação das saídas y e δ^4 e visualizada num ramo do tableau nas figuras abaixo .

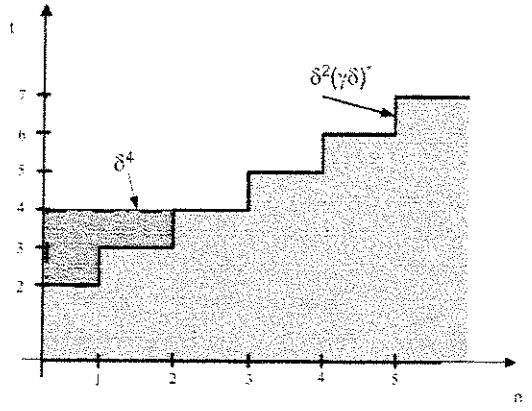


figura C.3

Especificação y $\rightarrow D^4u$ vista no plano

Arquivo de Entrada

Fórmulas correspondentes ao GET:

x1 <-> Du v GDx2 v G2Dx1

x2 <-> GDx1 v D2u

y <-> x1 v x2

Especificação:

y -> D4u

Arquivo de Saída

```
00
  F D4u
  F u
  F Gu
  T y
  T x2 x1
  T x2
  T D2u GDx1
  T GDx1
    *
01
  F D3u
  F u
  F Gu
  T Gx1
    *
02
  F D2u
  F u
  F Gu
    *
03
  F Du
  F u
  F Gu
    *
04
  F u
  F Gu
    *
10
  F u
  F Gu
  T Dx1
    *
11
  F u
  F Gu
  T x1
  T Du GDx2 G2Dx1
  T GDx2 G2Dx1
  T GDx2
    *
12
  F u
  F Gu
  T Gx2
    *
13
  F u
  F Gu
    *
14
  F u
  F Gu
    *
20
  F u
  F Gu
    *
21
  F u
  F Gu
  T Dx2
    *
22
  F u
  F Gu
  T x2
  T D2u GDx1
  T GDx1
    *
23
  F u
  F Gu
  T Gx1
    *
24
  F u
  F Gu
    *
30
  F u
  F Gu
    *
31
  F u
  F Gu
    *
32
  F u
  F Gu
  T Dx1
    *
33
  F u
  F Gu
  T x1
  T Du GDx2 G2Dx1
  T GDx2 G2Dx1
  T GDx2
    *
34
  F u
  F Gu
  T Gx2
    *
40
  F u
  F Gu
    *
41
  F u
  F Gu
    *
42
  F u
  F Gu
    *
43
  F u
  F Gu
  T Dx2
    *
44
  F u
  F Gu
  T x2
  T D2u GDx1
  T D2u
    *
45
  T Du
    *
46
  T u
    *
```

figura C.4

Ramo tableau para o exemplo 1

Exemplo 2

Modificando-se o GET do exemplo, como na figura C.5 pode-se comparar a trajetória de saída y com a trajetória de saída da transição $z = \delta(\gamma\delta^2)^*$ vistas na figura C.6. A especificação será dada por $z \rightarrow y$ O ramo aberto e terminado no mundo (i – 2, j – 5) da figura C.7 mostra que a especificação pode ser falseada.

Arquivo de Entrada

Fórmulas correspondentes ao GET:

x1 <-> Du v GDx2 v G2Dx1
x2 <-> GDx1 v D2u
y <-> x1 v x2
z<->Du v GD2z

Especificação:

z -> y

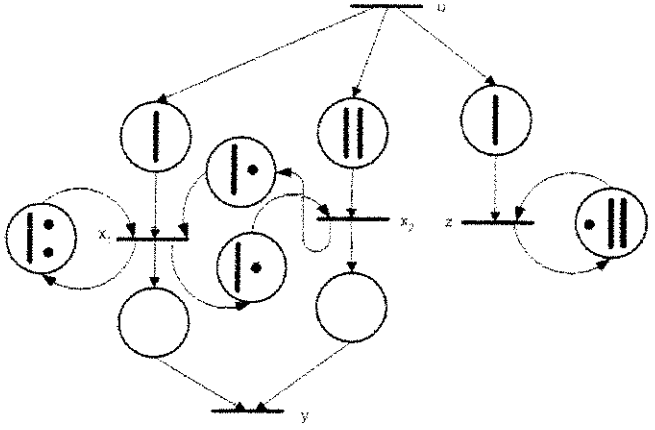


figura C.5
GET para o exemplo 2

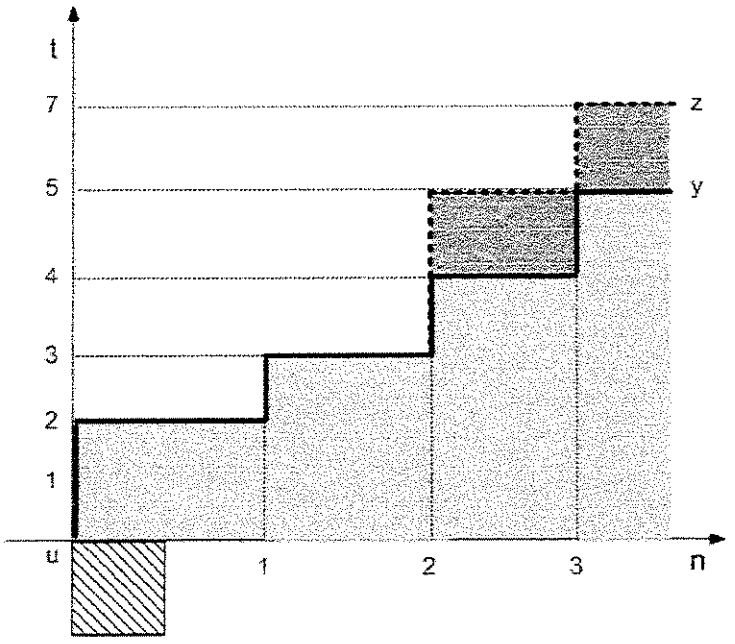
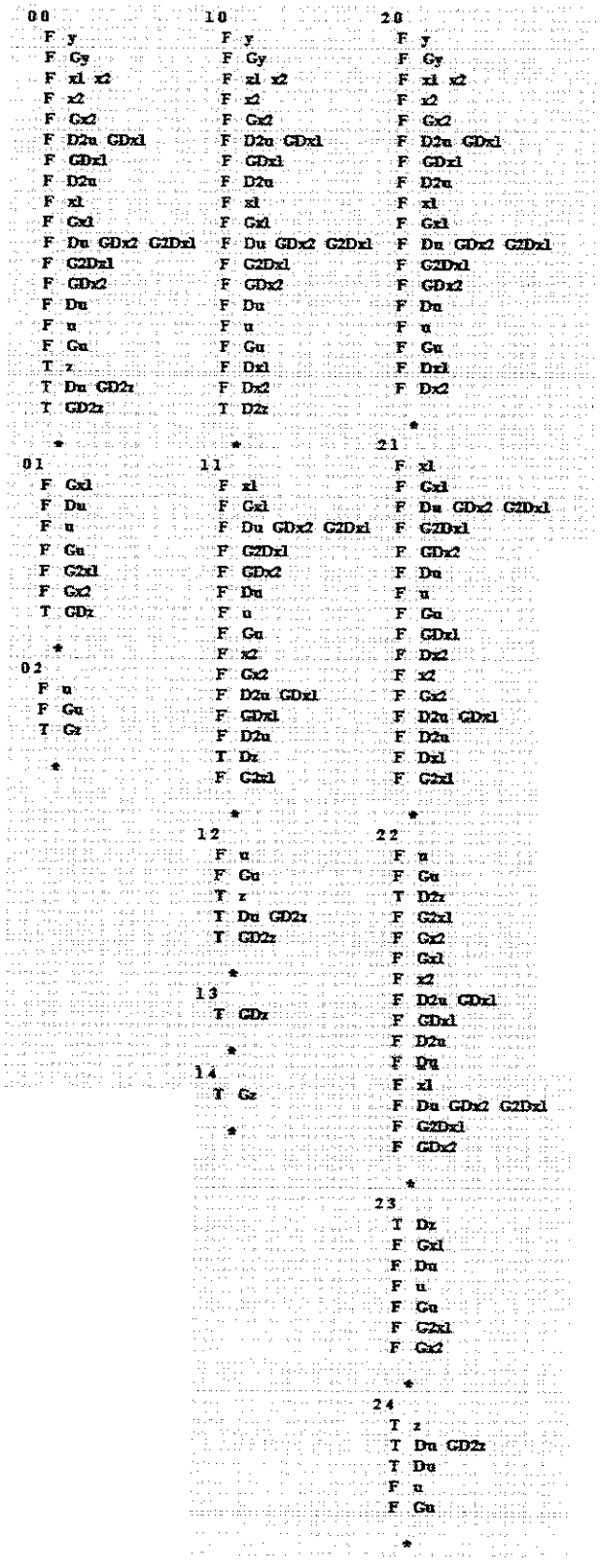


figura C.6

Trajetórias das transições y e z para o GET do exemplo 2

Arquivo de Saída


00
F y
F Gy
F xl x2
F x2
F Gx2
F D2u GDxl
F GDxl
F D2u
F xl
F Gxl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
F Du
F u
F Gu
T z
T Du GD2z
T GD2z
*
01
F Gxl
F Du
F u
F Gu
F G2xl
F Gx2
T GDx
*
02
F u
F Gu
T Gz
*
10
F y
F Gy
F xl x2
F x2
F Gx2
F D2u GDxl
F GDxl
F D2u
F xl
F Gxl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
F Du
F u
F Gu
F Dxl
F Dx2
T D2z
*
11
F xl
F Gxl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
F Du
F u
F Gu
F x2
F Gx2
F D2u GDxl
F GDxl
F D2u
T Dz
F G2xl
*
12
F u
F Gu
T z
T Du GD2z
T GD2z
*
13
T GDz
*
14
T Gz
*
20
F y
F Gy
F xl x2
F x2
F Gx2
F D2u GDxl
F GDxl
F D2u
F xl
F Gxl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
F Du
F u
F Gu
F Dxl
F Dx2
*
21
F xl
F Gxl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
F Du
F u
F Gu
F GDxl
F Dx2
F x2
F Gx2
F D2u GDxl
F D2u
F Dxl
F G2xl
*
22
F u
F Gu
T D2z
F G2xl
F Gx2
F Gxl
F x2
F D2u GDxl
F GDxl
F D2u
F Du
F xl
F Du GDx2 G2Dxl
F G2Dxl
F GDx2
*
23
T Dz
F Gxl
F Du
F u
F Gu
F G2xl
F Gx2
*
24
T z
T Du GD2z
T Du
F u
F Gu
*

Exemplo 3

No caso de uma especificação que compara as saídas do grafo dado pela figura abaixo.

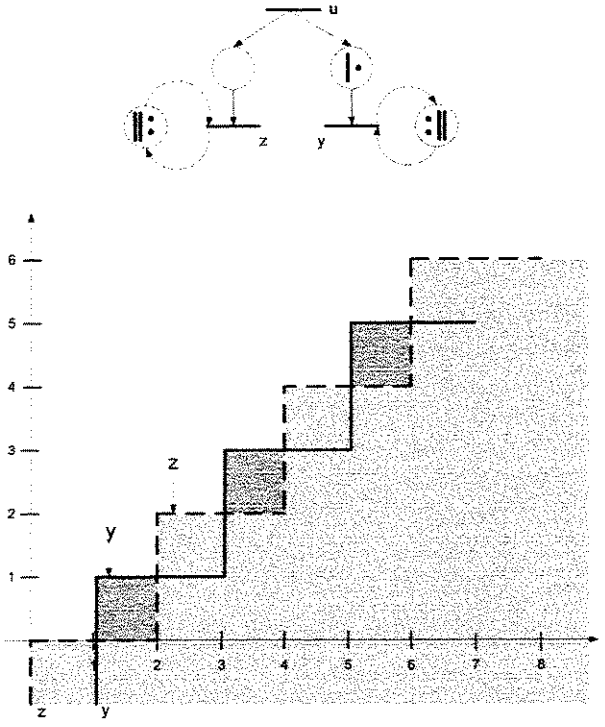


figura C.8
GET para o exemplo 3 e sua trajetória

Para y⟶z temos

Arquivo de Entrada

Fórmulas correspondentes ao GET:

z <-> G2D2z v u
y <-> G2D2y v GDu

Especificação:

z -> y

Arquivo de Saída

```
00
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  T z
  T GDu G2D2z
  T GDu

    *
01
  F G2Dy
  T Gu

    *
02
  F G2y

    *
10
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  T Du

    *
11
  F GDy
  T u
  F G2Dy

    *
```

figura C.9

Ramo terminado no mundo (i, j-1) para o exemplo 3

```
0 0
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  T z
  T GDu G2D2z
  T G2D2z
    *
0 1
  F G2Dy
  T G2Dz
    *
0 2
  F G2y
  T G2z
    *
```

```
1 0
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  T GD2z
    *
1 1
  F GDy
  T GDz
  F G2Dy
    *
1 2
  F Gy
  T Gz
  F G2y
    *
```

```
2 0
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  F D2y
  T D2z
    *
2 1
  F Dy
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  T Dz
  F GDy
  F G2Dy
    *
2 2
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  T z
  T GDu G2D2z
  T GDu
  F G2Dy
  F G2y
    *
2 3
  F G2Dy
  T Gu
  F G2y
    *
2 4
  F G2y
    *
```

```
3 0
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  F D2y
    *
3 1
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  F Dy
  F GDy
  F G2Dy
    *
3 2
  F y
  F Gy
  F u G2D2y
  F G2D2y
  F u
  F Gu
  F GD2y
  T Du
  F GDy
  F G2Dy
  F G2y
    *
3 3
  F GDy
  T u
  F Gy
  F G2Dy
  F G2y
    *
```

figura C.10

Ramo terminado no mundo (i–3, j–3) para o exemplo 3

Exemplo 4

Um outro tipo de especificação pode ter por objetivo comparar uma dada saída y, de um grafo com z = y^*. conforme pode-se ver na figura abaixo:

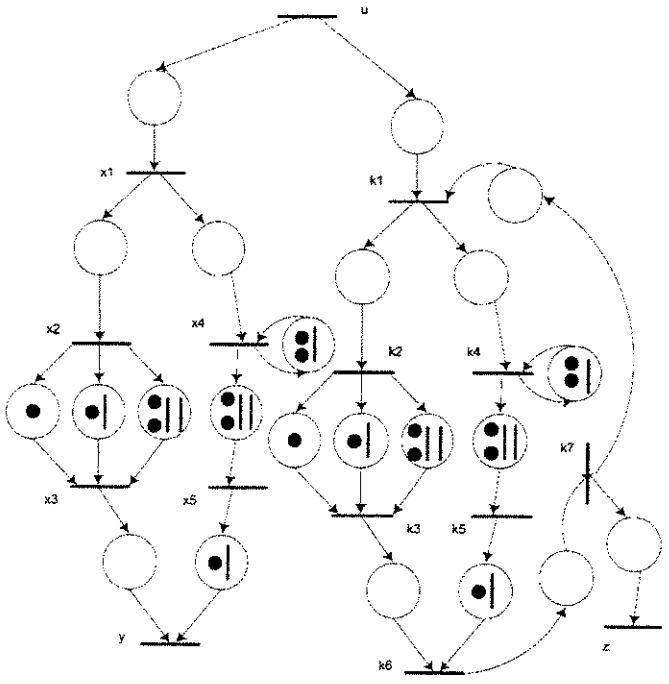


figura C.11
GET para o exemplo 4

Pela figura abaixo, que mostra as saídas têm-se que a partir do mundo (i-4,j-4) a especificação pode ser falseada.

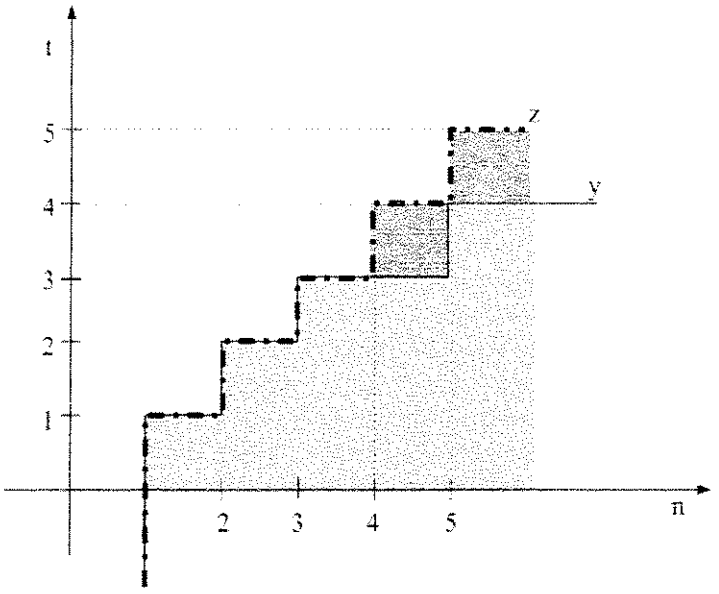


figura C.12

Trajetória no plano n×t para o GET do exemplo 4

O primeiro ramo terminado gerado pelo programa tableau conforme se vê no ramo abaixo mostra a terminação no mundo (i-4,j-4) .

Arquivo de Entrada

Fórmulas correspondentes ao GET:

z <-> k7

k7 <-> k6

k6 <-> k3 v GDk5

k5 <-> G2D2k4

k4 <-> G2Dk4 v k1

k1 <-> k7 v u

k3 <-> Gk2 v GDk2 v G2D2k2

k2 <-> k1

y <-> x3 v GDx5

x5 <-> G2D2x4

x4 <-> G2Dx4 v x1

x4 <-> u

x3 <-> Gx2 v GDx2 v G2D2x2

x2 <-> x1

Especificação:

z -> y

Arquivo de Saída

```
00
  F y
  F Gy
  F x3 GDx5
  F GDx5
  F x3
  F Gx3
  F Gx2 GDx2 G2D2x2
  F G2D2x2
  F GDx2
  F Gx2
  T z
  T k7
  T k6
  T GDk5 k3
  T GDk5
    *
01
  F Gx5
  F G2Dx2
  F Gx2
  T Gk5
    *
02
  F G2x2
    *
```

```
10
  F y
  F Gy
  F x3 GDx5
  F GDx5
  F x3
  F Gx3
  F Gx2 GDx2 G2D2x2
  F G2D2x2
  F GDx2
  F Gx2
  F Dx5
  F x5
  F Gx5
  F G2D2x4
  F GD2x2
  F Dx2
  F x2
  F x1
  F Gx1
  F u
  F Gu
  T Dk5
    *
11
  F x5
  F Gx5
  F G2D2x4
  F GDx2
  F Gx1
  F u
  F Gu
  T k2
  T k1
  T u k7
  T k7
  T k6
  T z
  T k3 GDk5
  T k3
  T GDk2 G2D2k2
  T GDk2
    *
12
  F Gx2
  F G2Dx4
  T Gk2
    *
13
  F G2x4
    *
```

```
20
  F y
  F Gy
  F x3 GDx5
  F GDx5
  F x3
  F Gx3
  F Gx2 GDx2 G2D2x2
  F G2D2x2
  F GDx2
  F Gx2
  F Dx5
  F x5
  F Gx5
  F G2D2x4
  F GD2x2
  F Dx2
  F x2
  F x1
  F Gx1
  F u
  F Gu
  F GD2x4
  F D2x2
    *
21
  F x5
  F Gx5
  F G2D2x4
22
  F x2
  F Gx2
  F x1
  F Gx1
  F u
  F Gu
  F GDx4
  T k2
  T k1
  T u k7
  T k7
  T k6
  T z
  T k3 GDk5
  T k3
  T GDk2 G2D2k2
  T GDk2
    *
23
  F Gx4
  T Gk2
    *
```

```
30
  F y
  F Gy
  F x3 GDx5
  F GDx5
  F x3
  F Gx3
  F Gx2 GDx2 G2D2x2
  F G2D2x2
  F GDx2
  F Gx2
  F Dx5
  F x5
  F Gx5
  F G2D2x4
  F GD2x2
  F Dx2
  F x2
  F x1
  F Gx1
  F u
  F Gu
  F GD2x4
  F D2x2
  F D2x4
  F x4
  F Gx4
  F x1 G2D2x4
    *
    *
33
  F x4
  F Gx4
  F x1 G2D2x4
  F G2D2x4
  F x5
  F Gx5
  F x1
  F Gx1
  F x2
  F u
  F Gu
  F Gx2
  T k2
  T k1
  T u k7
  T k7
  T k6
  T z
  T k3 GDk5
  T k3
  T GDk2 G2D2k2
  T GDk2
    *
34
  F G2Dx4
  T Gk2
```

```
40
  F x1
  F Gx1
  F x2
  F u
  F Gu
  F Gx2
    *
41
  F x1
  F Gx1
  F x2
  F u
  F Gu
  F Gx2
    *
42
  F u
  F Gu
  F x1
  F Gx1
  F x2
  F Gx2
    *
43
  F x4
  F Gx4
  F x1 G2D2x4
  F G2D2x4
  F x5
  F Gx5
  F x1
  F Gx1
  F x2
  F u
  F Gu
  F Gx2
  F GD2x4
  T Dk2
  F G2Dx4
    *
44
  F GDx4
  T k2
  T k1
  T u k7
  T u
  T x1
  T x2
  F G2Dx4
  F G2x4
    *
```

figura C 13

Exemplo 5

No caso de uma especificação que envolve um grafo com múltiplas entradas pode-se considerar que a entrada localizada em situação de terminação u_t = e, e que a(s) outra(s) entrada(s) que não aparecem no ramo ficam como $u_{\neg t} = \varepsilon$ conforme se pode ver nas figuras C.16 e C.17 mostrando os dois primeiros ramos abertos para a especificação z→y referentes ao grafo da figura C.14.

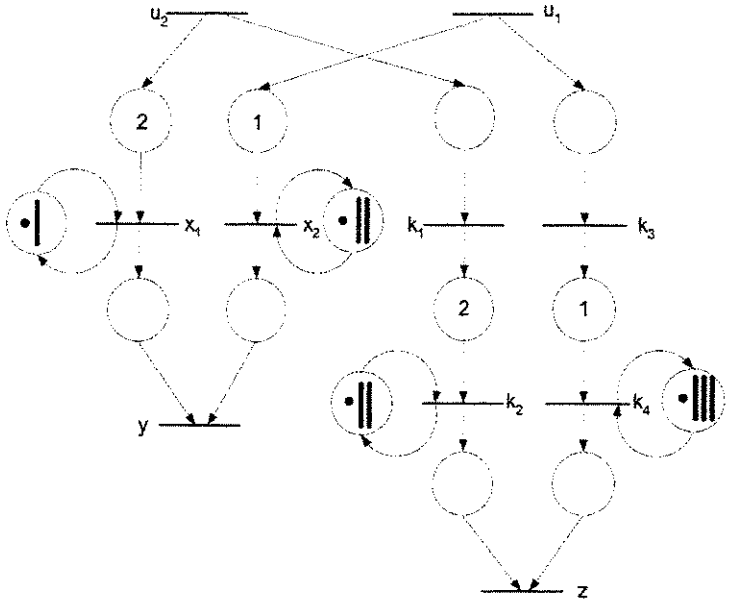


figura C.14
GET para o exemplo 5

A trajetória comparando z e y até o primeiro disparo, para ambas as entradas u_1 e u_2 pode ser vista abaixo.

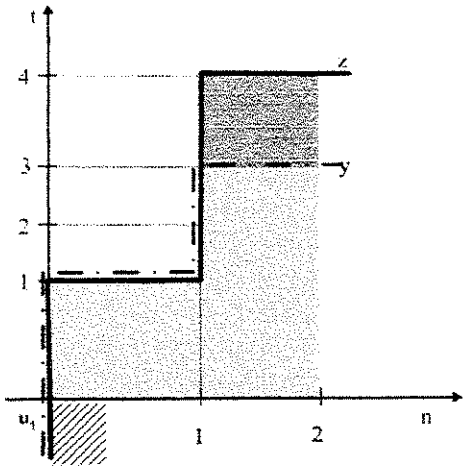


figura C.15

Trajetória no plano n×t para o GET do exemplo 5

Arquivo de Entrada

Fórmulas correspondentes ao GET:

z <-> k2 v k4

k4 <-> GD3k1 v Dk3

k3 <-> u1

k2 <-> GD2k2 v D2k1

k1 <-> u2

y <-> x1 v x2

x2 <-> GD2x2 v Du1

x1 <-> GDx1 v D2u2

Especificação:

z -> y

Arquivo de Saída

$$\begin{cases} u_1 = e \\ u_2 = \varepsilon \end{cases}$$

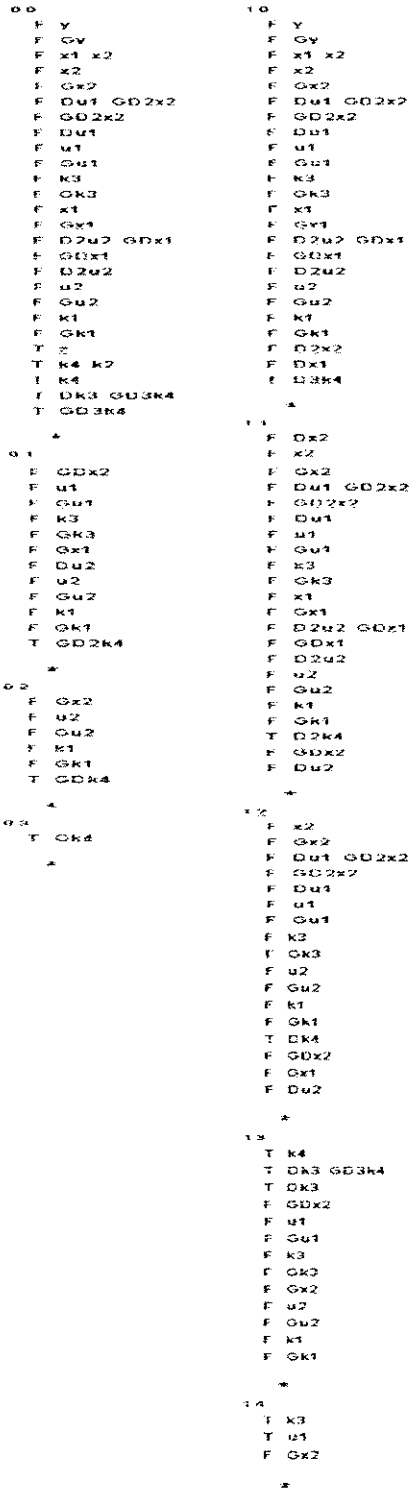


figura C.16

Ramo terminado em (i–1,j–4) para a variável de entrada ul

$$\begin{cases} u_1 = \varepsilon \\ u_2 = \varepsilon \end{cases}$$

```
0 0
  F y
  F Gy
  F x1 x2
  F x2
  F Gx2
  F Du1 GD2x2
  F GD2x2
  F Du1
  F u1
  F Gu1
  F k3
  F Gk3
  F x1
  F Gx1
  F D2u2 GDx1
  F GDx1
  F D2u2
  F u2
  F Gu2
  F k1
  F Gk1
  T z
  T k4 k2
  T k2
  T D2k1 GD2k2
  T GD2k2
    *
0 1
  F GDx2
  F u1
  F Gu1
  F k3
  F Gk3
  F Gx1
  F Du2
  F u2
  F Gu2
  F k1
  F Gk1
  T GDk2
    *
0 2
  F Gk2
  F u2
  F Gu2
  F k1
  F Gk1
  T Gk2
    *
1 0
  F y
  F Gy
  F x1 x2
  F x2
  F Gx2
  F Du1 GD2x2
  F GD2x2
  F Du1
  F u1
  F Gu1
  F k3
  F Gk3
  F x1
  F Gx1
  F D2u2 GDx1
  F GDx1
  F D2u2
  F u2
  F Gu2
  F k1
  F Gk1
  F D2x2
  F Dx1
  T D2k2
    *
1 1
  F Dx2
  F x2
  F Gx2
  F Du1 GD2x2
  F GD2x2
  F Du1
  F u1
  F Gu1
  F k3
  F Gk3
  F x1
  F Gx1
  F D2u2 GDx1
  F GDx1
  F D2u2
  F u2
  F Gu2
  F k1
  F Gk1
  T Dk2
  F GDx2
  F Du2
    *
1 2
  F x2
  F Gx2
  F Du1 GD2x2
  F GD2x2
  F Du1
  F u1
  F Gu1
  F k3
  F Gk3
  F u2
  F Gu2
  F k1
  F Gk1
  T k2
  T D2k1 GD2k2
  T D2k1
  F GDx2
  F Gx1
  F Du2
    *
1 3
  F GDx2
  F u1
  F Gu1
  F k3
  F Gk3
  T Dk1
  F Gx2
  F u2
  F Gu2
  F k1
  F Gk1
    *
1 4
  F Gx2
  T k1
  T u2
    *
```

figura C.17

Ramo terminado em (i=1,j=4) para a variável de entrada u2