Lógica Temporal Aplicada a Sistemas de Informação

Conference Paper · November 2007

DOI: 10.13140/RG.2.1.3595.5442

Silvia D. Rissino
Universidade Federal do Espírito Santo

Germano Lambert-Torres
PS Solutions, Itajuba, Brazil

VI Congress of Logic Applied to Technology
LAPTEC 2007
Santos / SP / BRAZIL - November 21-23, 2007
The Logic as Base in the Development and Application of New Technologies

PAPER 134

Lógica Temporal Aplicada a Sistemas de Informação

Silvia RISSINO[1], Germano LAMBERT-TORRES[2] e Helga G. MARTINS[2]

[1]Fundação Universidade Federal de Rondônia
BR 364, Km 9,5 – Porto Velho – 78900-500 – RO - Brasil
[2]Universidade Federal de Itajubá
Av. BPS 1303 – Itajubá – 37500-000 – MG – Brasil

Resumo: Este artigo é uma revisão da literatura, onde são apresentadas as potencialidades da utilização da Lógica Temporal em Sistemas de Informação, principalmente no que se refere ao desenvolvimento dos mesmos. O objetivo deste trabalho é mostrar como a Lógica Temporal é aplicada em sistema de informação, além de apresentar uma introdução elementar da lógica modal, da semântica de Kripke, a caracterização da Lógica Temporal e sua utilização em algumas áreas da Ciência da Computação.

Keywords: Lógica Temporal, Lógica Modal, Semântica de Kripke, Sistemas de Informação, Aplicações.

Introdução

O termo Lógica Temporal é usado para descrever um sistema de regras e simbolismos de representação do raciocínio, tendo a presença do tempo como elemento de destaque. O conceito de Lógica Temporal foi introduzido por volta dos anos de 1960 por Arthur Prior, com a designação de Lógica Tensa e conseqüentemente usada por lógicos e cientistas da computação [1].

Alguns sistemas lógicos são baseados na Lógica Temporal, como exemplo: Lógica computacional em árvore (CTL), Lógica para revestir tempo (LTL), Lógica Temporal de intervalos (ITL) e Lógica de tempo de ações (TLA). As mesmas incluem seu uso para formalizar o entendimento em assuntos filosóficos sobre tempo, definir as semânticas de expressões temporais em linguagem natural, para definir uma linguagem para codificar conhecimento temporal em inteligência artificial e atuar como uma ferramenta no controle de aspectos temporais da execução de programas.

Este artigo apresenta uma introdução elementar da lógica modal e da semântica de Kripke, além da caracterização da Lógica Temporal e sua utilização em algumas áreas da Ciência da Computação com ênfase em sistemas de informação.

1. Elementos da Lógica Modal

A lógica de primeira ordem tem limitação no sentido de não diferenciar entre o conceito de Possível e Necessário, pois de acordo com o formalismo clássico, fórmulas são apenas verdadeiras ou falsas, sem nenhum tipo de qualificação. A lógica modal permite representar o conceito de necessidade e possibilidade, ou seja, representa o estudo do comportamento das expressões de necessidade (é necessário que) e possibilidade (é possível que). Estes conceitos são formalizados a partir da noção de Mundo Possível, cuja interpretação pode ser descrita como uma alternativa imaginável ao mundo real. A lógica modal surge dentro do contexto da Lógica Temporal, pois a linguagem desta contém, além dos operadores verdade funcional, possui os operadores da lógica modal que são: Necessidade – notada por □; Possibilidade – notada por ◊.

A sintaxe e a semântica da lógica modal é derivada da lógica de primeira ordem. A Lógica Temporal especifica suas funções a partir da semântica da modal agregando o fator tempo [2]. Quantificadores da lógica modal - Necessidade □ e Possibilidade ◊:

$\Diamond\phi$: é possível que ϕ seja verdade;
$\Box\phi$: é necessário que ϕ seja verdade;
$\Box\phi\rightarrow\Diamond\phi$: tudo que é necessário é possível;
$\phi\rightarrow\Diamond\phi$: se algo é verdadeiro, então é possível;
$\phi\rightarrow\Box\Diamond\phi$: algo que é verdadeiro é necessariamente possível;
’ϕ→"’ϕ: tudo que é possível é necessariamente possível;

2. Semântica de Kripke

A semântica de Kripke é uma classe Kr de modelos de Kripke, onde o sistema K é o menor dos sistemas modais normais. Ou seja, ela é a interseção de todos os sistemas modais normais, justificado pelos seguintes princípios: trata-se de um sistema de lógica modal, com um conjunto de axiomas e regras de inferência que representam formalmente o raciocínio.

Seja P um conjunto de proposições atômicas, logo uma estrutura de Kripke é uma tupla (S, i , R, L) onde [3]:

- S é um conjunto finito de estados;
- $i \in S$ é o estado inicial;
- $R \subseteq S \times S$ é uma relação de transição total: $\forall s \in S\ .\ \exists s' \in S \cdot (s, s') \in R$
- $L : S \rightarrow$ □(P) é uma função que marca cada estado com o conjunto de fórmulas atômicas válidas nesse estado.

A semântica da Lógica Temporal pode ser definida sobre estruturas de Kripke, desta forma, as técnicas de especificação e verificação de propriedades podem ser apresentadas independentemente do formalismo do modelo a ser adotado.

3. Descrição da Lógica Temporal

A lógica clássica trata de uma espécie de “verdade verdadeira”, no sentido em que a mesma não dispõe de nenhum mecanismo que represente o tempo e suas conseqüências sobre os valores verdade nas fórmulas lógicas.

A Lógica Temporal possui como base a lógica modal e permite representar a variação dos estados do mundo em diferentes períodos de tempo, ou seja, permite verificar a veracidade das asserções ao longo do tempo. Já que uma asserção pode ser verdadeira em determinado instante de tempo e falsa em um outro instante.

Para os problemas de interesse da Inteligência Artificial e dos Sistemas de Informação, onde o tempo tem um papel importante, já que necessitam de uma representação dos eventos e sua seqüência no tempo para que possam ser eficientemente resolvidos [4]. Com exemplo, podem-se citar os problemas na área de diagnóstico médio, compreensão das “estórias médicas” ou mesmo problemas na engenharia de forma geral.

Na abordagem da Lógica Temporal as propriedades a serem verificadas são dadas através das fórmulas de Kripke (abordagem dos mundos possíveis) que representam um conjunto de estados, de transições entre estados e funções que rotulam cada estado com o conjunto de propriedades verdadeiras dentro dele.

A principal característica da Lógica Temporal é o fato de uma determinada fórmula lógica poder representar valores verdade diferentes em instantes distintos de tempo. Essa característica é formalizada através da introdução na sintaxe da linguagem lógica de primeira ordem de diversos operadores temporais. Neste caso, tem-se o Método de Argumentos Temporais, onde A é uma fórmula qualquer, P é um instante no passado, F é um instante no futuro, G todos os instantes no futuro e H todos os instantes do passado, conforme Tabela 1.

Tabela 1. Métodos de Argumentos Temporais para a Formula A.

PA	□t (t<now & *A* (t))	A foi verdade em algum instante do passado
FA	□t (now<t & *A* (t))	A será verdade em algum instante do futuro
GA	∀t (*A* do t<now → (t))	A será verdade em todos os instantes do futuro
HA	∀t (*A* do now<t → (t))	A foi verdade em todos os instantes do passado

Ao considerar a mudança dos valores verdades ao longo do tempo, as interpretações e modelos da Lógica Temporal, devem-se incluir uma estrutura que represente os instantes de tempo e sua relação de precedência.

Define-se a estrutura temporal como um conjunto vazio □ de instantes de tempo e uma relação de precedência $\prec$ sobre os elementos de □. Utilizando uma estrutura temporal é possível estender a função de associação □ e a semântica de operadores lógicos da lógica de primeira ordem, de forma que passem a depender de um instante de tempo t∈□. Para completar a semântica da Lógica Temporal basta adicionar as regras relativas aos operadores temporais:

- FA tem valor verdadeiro V no instante t∈□ se existe t’∈□ tal que t $\prec$ t’ e A tem valor V em t’;

- PA tem valor verdade V no instante $t \in$ □ se existir $t' \in$ □ tal que $t' \prec t$ e A tem valor V em t';

Os operadores G e H podem ser definidos através da relação de equivalência:

$GA \leftrightarrow \neg F \neg A$ e $HA \leftrightarrow \neg P \neg A$

Uma fórmula é Verdadeira na Lógica Temporal se ela representa valor verdade V em todos os instantes de tempo. Como na lógica modal, as propriedades da relação de precedência temporal determinam as características das diversas lógicas temporais. A relação de precedência sem nenhuma restrição corresponde à Lógica Temporal mínima K, que pode ser caracterizada pela regra de inferência *Modus Pones* e o seguinte conjunto de axiomas apresentados nas Figuras 1 e 2, onde são apresentadas as linearidades para frente e para trás.

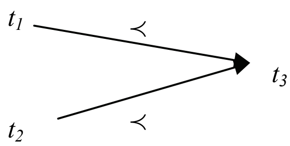

Figura 1 – Linearidade para frente.

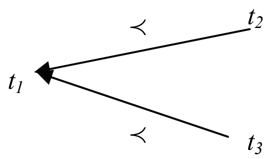

Figura 2 – Linearidade para trás.

Se A é uma tautologia, então:

$G(A \rightarrow B) \rightarrow (GA \rightarrow GB)$

$H(A \rightarrow B) \rightarrow (HA \rightarrow HB)$

$A \rightarrow HFA$

$A \rightarrow GPA$

Se A é um axioma, então $GA \wedge HA$.

A primeira restrição sobre a relação de precedência exige que o tempo seja Arborescente, isto é, que existia apenas um "passado", embora o futuro permaneça aberto. Esta restrição pode ser formalizada por:

$\forall t1, t2, t3$ □ □, $((t1 \prec t2) \wedge (t2 \prec t3)) \rightarrow t1$ □ $t3$

$\forall t1, t2, t3$ □ □, $((t1 \prec t2) \wedge (t2 \prec t3)) \rightarrow ((t1 \prec t2)$ □ $(t2 \prec t1)$ □ $(t1 = t2))$

A primeira condição é transitiva, devido à precedência no tempo. A segunda condição é chamada linear para trás, impedindo que exista instante de tempo relacionado, conforme apresentado na Figura 1. A Lógica Temporal associada a esta restrição, é caracterizada pelos axiomas de lógica K, além dos seguintes axiomas:

$FFA \rightarrow FA$

$(PA \wedge PB) \rightarrow P(A \wedge B)$ □ $P(A \wedge PB)$ □ $P(PA \wedge B)$

3.1 Natureza do tempo na Lógica Temporal

a) Linearidade do Tempo

Uma idéia bastante difundida sobre o tempo é que o mesmo seja uma seqüência linear de instantes de Arborescência. O tempo, no sentido clássico de Newton, é da forma linear. Logo, a restrição sobre a relação de precedência pode ser formalizada pela introdução de uma condição de Linearidade para Frente:

$$\forall t1, t2, t3 \square \square, ((t1 \prec t2) \Lambda (t1 \prec t3)) \rightarrow ((t2 \prec t3) \square (t3 \prec t2) \square (t2 = t3))$$

Esta condição impede que existam instantes de tempo relacionados conforme mostrado na figura 2. Os dois tipos de linearidade podem ser combinados em uma condição mais simples chamada conectividade:

$$\forall t1, t2 \square \square, ((t1 \prec t2) \square (t2 \prec t1) \square (t1 = t2))$$

Esta nova Lógica Temporal, chamada de K1, pode ser caracterizada pelos axiomas da lógica Kb, mais o seguinte axioma:

$$(FA \Lambda FB) \rightarrow F(A \Lambda B) \square F (A \Lambda FB) \square F(FA \Lambda B)$$

b) Limites do Tempo – Início e Fim

Outra situação importante sobre a natureza do tempo é a existência dos seus limites, isto é, a existência ou não de um tempo inicial e de um tempo final. A física clássica assume que o tempo é ilimitado tanto com relação ao passado como em relação ao futuro, sendo este pensamento formalizado pelas condições sobre a relação de precedência:

$$\forall t2 \square \square, \exists t1 \square \square, t1 \prec t2$$

$$\forall t1 \square \square, \exists t2 \square \square, t2 \prec t1$$

Esta nova Lógica Temporal, chamada de Ks, pode ser caracterizada pelos axiomas da Lógica Kl mais os seguintes axiomas:

$$GA \rightarrow FA$$

$$HA \rightarrow PA$$

c) Densidade do Tempo

Uma série linear pode apresentar a estrutura dos números inteiros, racionais ou reais. A lógica de Ks admite uma estrutura como a dos números inteiros, onde podem existir dois instantes sem que exista um terceiro que seja intermediário aos instantes limites.

Um tempo denso como a estrutura dos números racionais onde sempre existe um instante intermediário entre dois outros, pode ser formalizado através da condição:

$$\forall t1, t2 \square \square, \exists t3 \square \square ((t1 \prec t2) \rightarrow (t1 \prec t3) \Lambda (t3 \prec t2))$$

A lógica que adota esta restrição, chamada Kp, pode ser caracterizada pelos axiomas da lógica Ks mais o seguinte axioma:

$$FA \rightarrow FFA$$

Um tempo contínuo com a estrutura dos números reais, pode ser caracterizada da seguinte forma: Se dividir o conjunto □ em dois outros conjuntos □1 e □2, tais que □ = □1 ∪ □2 e qualquer elemento de □1 precedem todos os elementos de □2, então sempre deve existir um elemento de □ que suceda todos os elementos de □1 e preceda todos os elementos de □2. Esta condição pode ser formalizada por:

$$\forall \square , \square \subseteq \square , ((\square = \square 1 \cup \square 2) \Lambda (\forall t1 \in \square 1, \forall t2 \in \square 2, t1 \prec t2)) \rightarrow$$

$$(\exists t \in \square, \forall t1 \in \square 1, \forall t2 \in \square 2, (t1 \prec t)) \Lambda (t \prec t2))$$

Esta condição é absorvida pelo seguinte axioma: ((GA → PGA) Λ G(GA → PGA) Λ H(GA → PGA)) → (GA →HA), que origina, juntamente com os axiomas da lógica Kp, à Lógica Temporal contínua, chamada Kc.

3.2 Forma de Representar o Tempo na Lógica Temporal

As lógicas temporais (CTL, LTL, ITL e TLA) divergem quanto ao modo como representam o tempo, existindo dois modelos básicos para sua representação [3]:

a) Tempo Linear: O comportamento do sistema consiste no conjunto de traços infinitos que começam no estado inicial *i*;

b) Tempo Ramificado: Todo o comportamento do sistema é capturado por uma árvore de computação de profundidade infinita cuja raiz é o estado inicial *i*.

Ambos os modelos podem ser calculados a partir da estrutura de Kripke, mas o ramificado tem mais informações, sendo que, existem propriedades que apenas podem ser verificadas usando tempo ramificado.

Na lógica linear temporal (*linear temporal logic* – LTL), as fórmulas são interpretadas sobre traços infinitos, além dos conectivos usuais da lógica proposicional temos os seguintes operadores temporais utilizados, no caso de tempo ramificado: *Next* X f quando f é válida no próximo estado; *Future* F f quando f é eventualmente válida; *Globally* G f quando f é sempre válida; *Until* f U g quando f é válida até que g o seja; *Release* f R g quando a ocorrência de um estado onde f é válida liberta g de o ser.

4. Lógica Temporal e suas Aplicações

As aplicações da Lógica Temporal em ciência da computação são apresentadas nos trabalhos de renomados pesquisadores com C. Caleiro, Michel Fisher, Chiara Ghidini e outros. No final da década de 1970, o estilo modal da Lógica Temporal encontrou a aplicação extensiva na área de informática. Sendo aplicada e relacionada à especificação e verificação dos programas, em especial naqueles com execução simultânea, isto é, o processamento é executado por dois ou mais processadores, que trabalham em paralelo, com o objetivo de assegurar o comportamento correto dos programas, sendo necessário especificar a maneira em que as ações dos vários processadores são relacionadas [5]. O sincronismo relativo das ações deve ser coordenado com muito cuidado para assegurar que a integridade da informação compartilhada entre os processadores esteja mantida. Entre as noções chaves está a diferença entre as propriedades do "*liveness*" das propriedades lógico-temporal, onde Fp assegura que os estados desejáveis sejam obtidos no curso da execução, e o Gp que assegura que os estados indesejáveis nunca sejam alcançados.

A Lógica Temporal pode ser também encontrada em aplicações de inteligência artificial, como a construção de agentes, sendo um desses tipos os agentes BDI (*Belief, Desire and Intention*), que possuem uma arquitetura de crenças, desejos e intenções que necessitam formalização para que tenham real utilidade. A formalização de tais termos pode ser realizada através de sua representação lógica. Geralmente, utiliza-se lógica modal para essas situações, mas nos anos 1990, o formalismo lógico baseado em Lógica Temporal passou a ser utilizada, já que a mesma permite a análise de como a crença do agente futuro pode afetar os desejos e intenções do presente [6].

Outra aplicação da Lógica Temporal é na engenharia de software, através do uso de técnicas de inteligência artificial, pois esta ultima utiliza Lógica Temporal na melhoria dos ambientes de construção e qualidade de software. Exemplo na engenharia de software é o ambiente de desenvolvimento PANDORA [7], que é uma máquina de processos baseada em conceitos de Lógica Temporal que utiliza a linguagem PROLOG na sua programação. Esse ambiente possui um algoritmo de aplicação de regras que otimiza os passos da execução. Neste caso, as regras são implementadas em Lógica Temporal, o que permite expressar e estabelecer quais atividades é permitido a cada momento, além de como as mesmas são sincronizadas, sendo que o tempo é caracterizado por uma linha seqüencial simples de eventos.

A Lógica Temporal surge como uma ferramenta para o desenvolvimento dos sistemas de informação, já que os mesmos são compostos por módulos relacionados de forma hierárquica entre si. A Lógica Temporal é utilizada nos componentes do sistema relacionados principalmente na construção dos programas de forma geral, no armazenamento e consulta aos dados. No nível mais próximo ao usuário, há a interface homem-máquina; no nível intermediário, o sistema de informação deve ter mecanismos de processamento de dados para a entrada, edição, análise, visualização e saída dos dados; e no nível mais interno do sistema, deve ter um sistema de gerência de bancos de dados, A figura 3 apresenta uma proposta esquemática de uma arquitetura geral de sistemas de informação.

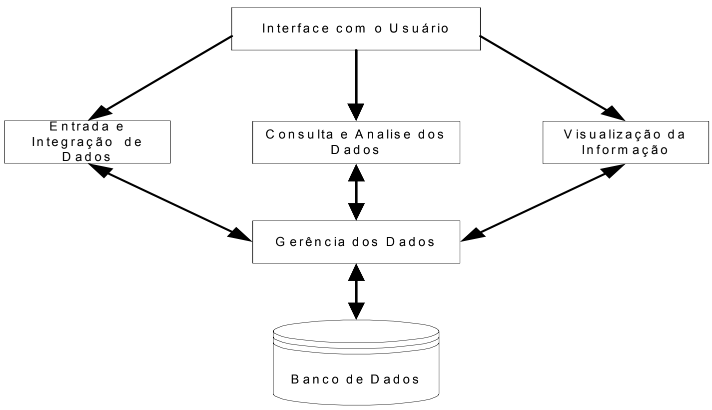

Figura 3 – Arquitetura de Sistemas de Informação

Na implementação dos bancos de dados históricos, a Lógica Temporal é utilizada na elaboração das regras de restrição de integridade e nas consultas aos dados temporais. Utiliza-se a Lógica Temporal, principalmente, em função do modelo de dados empregado, pois a mesma é utilizada na elaboração das expressões de consultas e implementadas em linguagem correspondente ao modelo empregado. Para obter sucesso na consulta em banco de dados temporal é necessário que as linguagens de consultas possam manipular a dimensão tempo embutida nos dados, através do qual se possa inferir, sobre as informações que não estão explicitamente armazenadas. Isto significa dizer que, o resultado das consultas pode ser mensurado em função da integridade das expressões lógicas usadas.

Esta revisão da literatura faz parte de um estudo preliminar sobre a utilização da Lógica Temporal em banco de dados históricos, principalmente no que se refere a garantia da recuperação da informação para tomada de decisão nos casos onde a utilização da evolução dos dados se faz necessária.

5. Conclusão

Este artigo apresentou as potencialidades da utilização da Lógica Temporal em Sistemas de Informação, principalmente no que se refere ao desenvolvimento dos mesmos. A Lógica Temporal é utilizada para a especificação e verificação de programas, principalmente os que possuem a caracterisitica de execução simultânea. Na engenharia de software, o uso da Lógica Temporal se faz através do uso de técnicas de IA, sendo que um dos objetivos é a melhoria dos ambientes de construção e qualidade de software.

O uso de Lógica Temporal em sistemas de informação é enfatizado principalmente no módulo de armazenamento de dados, pois o banco de dados é o módulo que corresponde ao núcleo dos sistemas de informação, já que é nele que os dados históricos são armazenados. Atualmente, as aplicações estão utilizando Lógica Temporal para resolvem muitos problemas da engenharia, da medicina, de grandes bases de dados como as de seqüência genética entre outros. A Lógica Temporal apresenta-se como uma alternativa para o tratamento de dados e recuperação de informações históricas. Isto se deve à sua principal característica, ou seja, uma determinada fórmula lógica poder representar valores verdades diferentes em instantes distintos de tempo. A utilização da Lógica Temporal em sistemas de informações é de essencial importância, principalmente naqueles em que a evolução dos dados no decorrer do tempo é fator de confiabilidade e integridade.

7. Referências

[1] Copeland, B. Jack. Arthur Prior. First published Mon Oct 7, 1996; substantive revision Sat Aug 18, 2007 Disponível em http://plato.stanford.edu/entries/prior/. Acessado em 20/07/2007.

[2] Garson, James. Modal Logic. First published Tue Feb 29, 2000; substantive revision Sat May 5, 2007. Disponível em http://plato.stanford.edu/entries/logic-modal/. Acessado em 30/07/2007.

[3] Cunha, Manuel Alcino. Lógica Temporal. Junho 2005. Disponível http://wiki.di.uminho.pt/twiki/pub/Education/PeC/MaterialApoio/temporal.pdf. Acessado em 05/05/2007.

[4] Bittencourt, Guilherme. Inteligência Artificial: Teorias e Ferramentas. Campinas: Instituto de Computação, UNICAMP, 1996. p.75-94.

[5] Caleiro, C.; Saake, G.; Sernadas, A.. Deriving Liviness Goals From Temporal Logic Specifications. Academic Press Limited. 1996. Disponível em http://citeseer.ist.psu.edu/cache/papers/cs/caleiro96deriving.pdf. Acessado em 20/07/2007.

[6] Fisher, Michael; Ghidini, Chiara. The ABC of Rational Agent Modelling. AAMAS'02, July 15-19, 2002, Bologna, Italy. Disponível em delivery.acm.org/10.1145/550000/544943/p849-fisher.pdf. Acessado em 10/06/2007.

[7] Lago, P.; Manalti, G. Pandora: a temporal logic based process engine. Workshop in logic programming applied to software engineering. 1994. Disponível em http://citeseer.ist.psu.edu/cache/papers/cs/3391/clp94.pdf/pandora-a-temporal-logic.pdf. Acessado em 28/07/2007.