Uma Abordagem de Autômatos Adaptativos Usando Teoria das Categorias

J. L. R. Monteiro e J. E. Kögler Jr., *Senior Member IEEE*

***Resumo* — A teoria das categorias oferece em uma abordagem unificada aspectos de topologia e da álgebra. Seu poder de abstração permite abstrair de maneira prática o tratamento de sistemas e processos de naturezas bastante genéricas. Apresenta-se como potencial benefício ao estudo do formalismo de autômatos. Introduz-se neste trabalho uma proposta para uma abordagem do citado formalismo para autômatos adaptativos.**

***Palavras-chave* — Modelos matemáticos, Teoria das Categorias, Autômatos, Autômatos Adaptativos.**

I. Introdução

A Teoria das Categorias é relativamente recente (apresentada pela primeira vez por S. Mac Lane em 1945, publicada em [1]). Ela facilita o tratamento simultâneo de aspectos topológicos e algébricos. Essa teoria trata de forma conjunta estruturas matemáticas genéricas e relacionamentos entre elas.

Ela fornece uma descrição abstrata de sistemas, estruturas, operações e processos, desta forma se constituindo em um jargão e um ambiente consistente e unificado para o estudo de diversas áreas sob o ponto de vista da matemática. Sua capacidade de generalização, abstração e unificação é o grande mérito de Teoria das Categorias. Suas aplicações para a Ciência da Computação são inúmeras, como por exemplo a Semântica Categorial [2] e a Lógica Categorial[3][4].

Essa teoria permite trabalhar de forma concisa e expressiva diversos conceitos complexos como o formalismo de autômatos adaptativos. Neste trabalho propõe-se um modelo para a o formalismo adaptativo[5] de autômatos, expresso sob essa nova ótica.

Na próxima seção apresentamos os conceitos fundamentais, proporcionando uma visão de em larga escala da teoria das categorias. Na seção 3 ilustra-se a aplicação da teoria para representar, primeiramente, os autômatos finitos, que servirá de inspiração para a próxima construção. Em seguida, na seção 4, revemos a definição formal do autômato adaptativo[6], introduzindo-se uma proposta de modelo via teoria das categorias. Finalmente, na seção 5 faz-se uma breve exploração das conseqüências dessa nova teoria ao paradigma adaptativo.

II. Conceitos da Teoria das Categorias

A Teoria das Categorias é, sob certa forma, uma generalização da Teoria dos Conjuntos, embora seu alcance seja muito maior. Conjuntos constituem coleções de objetos que satisfazem a uma dada especificação. As categorias são coleções de objetos e, simulatâneamente, das relações entre eles, denominadas *morfismos*. Um exemplo particular de categoria, é a denominada categoria **Set**, cujos objetos são conjuntos e os morfismos são funções entre cada par de conjuntos. Pode-se criar uma categoria **Aut** tendo autômatos finitos como objetos, dotadas de morfismos que são mapeamentos entre cada dois autômatos.

A seguir serão apresentadas algumas propriedades das categorias.

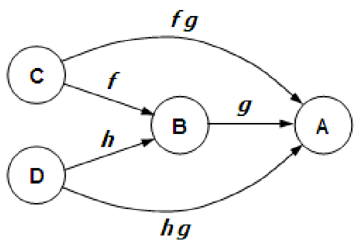

Fig. 1: Composição associativa: nessa figura a composição dos morfismos *f* com *g* resulta em *fg*.

A. Propriedades Fundamentais

Uma categoria C é definida por:

- Conjunto Obj(C) dos objetos da categoria;
- Conjunto Mor(C) dos morfismos da categoria, que interrelacionam os elementos de Obj(C);
- Uma operação de composição associativa em Mor(C) x Mor(C), que resulta no morfismo composto entre dois objetos adjacentes a um terceiro (ilustrado na Fig. 1);
- Uma operação de identidade, garantindo que exista um morfismo de cada objeto consigo mesmo.

J.L.R. Monteiro é estudante de doutorado em Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo (e-mail.: jmonteiro@gmail.com).

J.E. Kögler Jr. é pesquisador doutor do Departamento de Engenharia de Sistemas Eletrônicos da Escola Politécnica da Universidade de São Paulo, Av. Prof. Luciano Gualberto, 158, Trav.3 05508-900 São Paulo, SP, Brasil (e-mail.: kogler@lsi.usp.br).

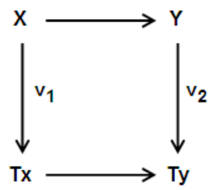

Fig. 2: Diagrama comutativo: nesse diagrama são expressos morfismos comutativos entre *X* e *Tx* e *Y* e *Ty*.

Uma das formas de se exprimir as propriedades em Teoria das Categorias é através de diagramas. Se a composição de todos os caminhos entre dois objetos do diagrama for equivalente, o diagrama é dito *comutativo* (como na Fig. 2). Os diagramas são recursos freqüentemente utilizados na demonstração de teoremas envolvendo categorias.

Uma outra propriedade importante da teoria é a dualidade, através da qual pode-se obter uma categoria inversa ou oposta, bastando inverter o sentido dos seus morfismos. Demonstra-se que as propriedades válidas em uma categoria C são igualmente válidas para a sua categoria dual, representada por C^{op} (exemplificado na Fig. 3).

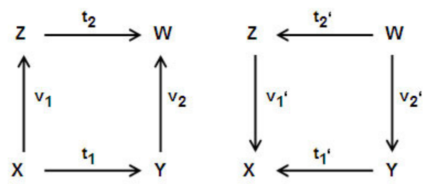

Fig. 3: Categorias duais: a categoria representada do lado direito é a dual à categoria do lado esquerdo.

B. Funtores

Funtores são mapeamentos entre categorias, que preservem a estrutura da categoria domínio na categoria co-domínio, através do mapa. Um funtor *F* (dito *covariante*), da categoria C para a categoria D apresenta as características construtivas:

1. associa para cada objeto X de Obj(C) um objeto F(X) em Obj(D);
2. associa para cada morfismo f : X → Y de Mor(C), um morfismo F(f) : F(X) → F(Y) em Mor(D)

tal que, sendo g de Mor(C), as seguintes propriedades valem:

- $F(id_X) = id_{F(X)}$
- F(g o f) = F(g) **o** F(f) para todos os morfismos f:X → Y e g:Y → Z.

com **o** indicando a composição entre funtores.

C. Tipos especiais de Funtores

Existem diversos tipos notáveis de funtores, destacando-se em especial os seguintes, que utilizaremos em nossas construções posteriores:

Funtor Esquecimento: é um funtor que transforma uma categoria em outra, "esquecendo" parte da estrutura (Fig.4). Por exemplo, o funtor *F: Vect* → *Set*, leva a categoria de todos os espaços vetoriais na categoria dos conjuntos esquecendo a estrutura de espaço vetorial.;

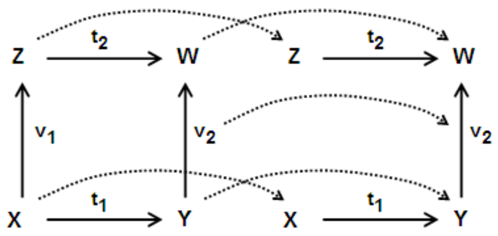

Fig. 4: Funtor Esquecimento: Transforma uma categoria, removendo-lhe parte da estrutura.

Funtor Livre: é o funtor dual do funtor esquecimento, levando cada objeto ao grupo livremente gerado pelo mesmo objeto, permitindo incluir novas estruturas de relacionamento (morfismos) entre os objetos (Fig. 5).

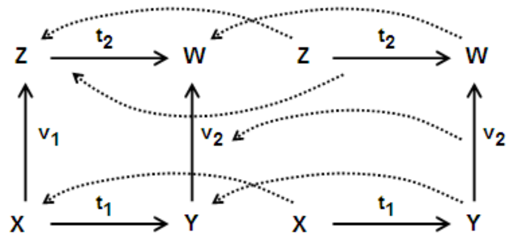

Fig. 5: Funtor Livre: transforma uma categoria, adicionando nova estrutura.

III. Modelagem do Autômato Finito

Primeiramente apresentaremos um modelo dos autômatos finitos usando a teoria das categorias, o que auxiliará na compreensão de um modelo posterior para autômatos adaptativos.

Formalmente, os autômatos finitos podem ser especificados pelas 6-uplas:

A = (I, O, S, s0, t, p) , na qual:

- *I* é o conjunto de símbolos de entrada;
- *O* é o conjunto de símbolos de saída;
- *S* representa o conjunto de estados;
- *s0* é o estado inicial;
- *t: I x S* → *S* representa a função de transições;
- *p: I x S* → *O* representa a função de saída.

Sejam *A* e *A'* dois automatos finitos. Um morfismo *f* entre *A* e *A'* escreve-se como:

f = (fI,fO,fS) : A → *A'*, tal que:

fS(s0) = s0' (preservação do estado inicial)

fS(t(i,s)) = t'(fI(i),fS(s)) (preservação das transições)

fO(p(i,s)) = p'(fI(i),fS(s)) (preservação das saídas)

A categoria ***Aut*** dos autômatos finitos pode ser escrita como [7]:

- *Aut=(Obj(Aut), Mor(Aut), Comp(Aut), Ident(Aut))*
- *Obj(Aut): A = (I, O, S, s0, t, p)*
- *Mor(Aut): f = (fI,fO,fS) : A → A'*
- *Comp(Aut)*: a composição é associativa, pois as componentes do morfismo operam nos conjuntos *I, O, S;*
- *Ident(Aut)*: a identidade se dá pelo morfismo que preserva a identidade nos conjuntos *I, O, S.*

Como a categoria ***Aut*** é definida dentro de conjuntos, é relativamente simples observar que a composição de morfismos entre autômatos finitos é associativa, e que a identidade é preservada [7].

IV. Formalizando o Autômato Adaptativo

Com base no entendimento da aplicação da teoria para o autômato finito, explorado na seção anterior, pode-se similarmente formalizar os autômatos adaptativos através da teoria das categorias.

Os autômatos finitos adaptativos são especificados pela 8-upla abaixo [5]:

ADk = (I, O, Sk, sk, tk, p, BA, AA) , na qual:

- *Sk* é o conjunto dos estados, na iteração *k*;
- *s0* é o estado inicial;
- *tk: AA x I x S x BA → S* é a função de transições adaptativa;
- *p: I x S → O* representa a função de saída;
- *BA*: conjunto das funções adaptativas que ocorrem antes;
- *AA*: conjunto das funções adaptativas que ocorrem depois.

Note que as várias iterações dos autômatos adaptativos pressupõem uma indexação através de índices k. Isso se faz necessário devido ao próprio conceito de adaptação, que implica em mudanças estruturais acontecendo de uma situação (iteração) para a seguinte.

Introduziremos agora a categoria ***AAut*** dos autômatos adaptativos. Ela se apresenta como uma coleção de autômatos adaptativos e de morfismos entre eles. Note que ela deve ser uma supercategoria sobre a categoria de ***Aut***. Isto é, cada autômato adaptativo componente de ***AAut*** deve ser visto individualmente como uma categoria menor, tipo ***Aut***, em que seus elementos constituem as diversas iterações compatíveis com a adaptação de um dado autômato aplicando-se suas funções adaptativas convenientemente. Então, os morfismos entre os objetos de ***AAut*** são funtores entre autômatos de ***Aut***:

- *AAut=(Obj(AAut),Mor(AAut),Comp(AAut), Ident(AAut))*
- *Obj(AAut):AA = {ADk, | ADk = (I, O, Sk, sk, tk, p, BA, AA) }*
- *Mor(AAut):F : AA → AA'*
- *Comp(AAut)*: a composição associativa de funtores;
- *Ident(AAut)*: funtor identidade.

Os funtores acima propostos devem levar autômatos adaptativos em autômatos adaptativos, mapeando adequadamente os elementos homólogos. Isso requer que funções adaptativas sejam mapeadas em funções adaptativas. Entre as dificuldades técnicas que se tem de resolver nesse cenário, encontra-se o caso em que duas classes de topologias distintas devem ser relacionadas. Por exemplo, a remoção de estados e transições pode ser considerada através da aplicação de funtores tipo esquecimento. Portanto embora internamente cada categoria ***Aut*** deva satisfazer às condições a ela anteriormente impostas, a categoria ***AAut*** não precisa satisfazê-las, podendo ser bem mais geral. Da mesma forma, pode-se introduzir novos estados e transições, o que se consegue através de funtores do tipo livre, adicionando estados e transições entre objetos de ***AAut***.

Um dos benefícios desse tipo de abordagem é a investigação de otimização estrutural, conseguida através de construtos comuns em teoria das categorias que conduzem a generalizações, como produto e limites e seus duais, os coprodutos e os colimites [1]. Por exemplo, poder-se-ia investigar qual a realização mínima de um dado autômato adaptativo que satisfaz a uma determinada restrição.

V. Conclusões

De uma forma geral, o formalismo de categorias permite formalizar adequadamente os autômatos adaptativos, porém ainda é preciso demonstrar formalmente que a especificação proposta satisfaz às condições construtivas de uma categoria, o que se constitui em um trabalho em andamento. Todavia, por ser uma supercategoria, ***AAut*** herda naturalmente as propriedades construtivas das categorias, o que por si só constiui uma pré-prova. Talvez o aspecto mais difícil se encontre na determinação de funtores para aplicações a situações de particular interesse e no estabelecimento de teoremas que caracterizem os princípios de otimização, por exemplo, aplicáveis à determinação das realizações mínimas.

A abordagem de autômatos adaptativos via teoria das categorias possibilita também sua potencial adoção na realização de estruturas evolutivas que caracterizam processos cognitivos [8]. Realizações como o MES (Memory Evolutive System) [9][10][11], trazem um poder de representação de processos cognitivos fundamentado na teoria de categorias. Nesse cenário, um aspecto interessante de se explorar constitui-se no estudo de como a indexação utilizada na estruturação de cada objeto da supercategoria ***AAut*** pode ser relacionada com a indexação de um outro objeto, criando-se um modelo de temporização local interna, característico de sistemas pulsados assíncronos [9].

REFERÊNCIAS

[1] Mac Lane, S. Categories for the Working Mathematician. Springer. ISBN 0-38-798403-8, 1972.

[2] Pitts, A.M. Categorical Logic. Chapter 2 of S. Abramsky and D. M. Gabbay and T. S. E. Maibaum (Eds) Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures, Oxford University Press, 2000.

[3] Lambek, J. & Scott, P.J. Introduction to Higher Order Categorical Logic, Cambridge University Press, Cambridge, UK, 1986.

[4] Barr, M. and Wells, C. Category Theory for Computing Science, Hemel Hempstead, UK, 1990.

[5] Neto, J. J. Adaptive Rule-Driven Devices - General Formulation and Case Study. Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Vol.2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.

[6] Neto, J. J. & Pariente, C. A. B. Adaptive Automata - a Revisited Proposal. Lecture Notes in Computer Science. J.M. Champarnaud, D. Maurel (Eds.): Implementation and Application of Automata 7th International Conference, CIAA 2002, 2002, Vol.2608, Tours, France, July 3-5, Springer-Verlag, 2002, pp. 158-168

[7] Martini, A.; Ehrig, H. & Nunes, D. Elements of basic category theory. Technical Report 96-5, Technical University Berlin, Março 1996.

[8] Kogler Jr.,J.E. & Inojosa,R. – First Steps Towards a Cognitive Architecture Based on Adaptive Automata. Proceedings of the Workshop on Modelling Adaptive and Cognitive Systems – ADAPCOG – Salvador, Brazil, October 30, 2008

[9] Ehresmann, A.; Vanbremeersch, J.-P. - *Memory Evolutive Systems – Hierarchy, Emergence, Cognition*, Studies in Multidisciplinarity V4, Elsevier, Amsterdam, 2007

[10] Monteiro, J.L.R., Ribeiro, J.H.R., Kogler Jr., J.E., Netto, M.L. – On the Requirements for Simulating a Memory Evolutive System – Proceeedings of Brain Inspired Cognitive Systems, BICS, São Luiz, Brazil,June 24-27, 2008

[11] Monteiro, J.L.R., Kogler Jr., J.E., Ribeiro, J.H.R., Netto, M.L. – On Building a Memory Evolutive System for Application to Learning and Cognition Modelling – submitted - Brain Inspired Cognitive Systems – Ed. A. Hussein and V Ctsuridis – Springer – 2009 – To appear.

Julio Lima do Rego Monteiro . Bacharel em Ciências Moleculares pela Universidade de São Paulo (1999). Mestre em Engenharia Elétrica – Sistemas Digitais, pela Universidade de São Paulo (2004) com dissertação sobre inteligência artificial, sistemas multiagentes e simulação social. Atualmente está cursando o doutorado em Engenharia Elétrica, também na Universidade de São Paulo, estudando a simulação de propriedades e processos cognitivos para resolver problemas relacionados com a modelagem computacional da memória, fundamentado na teoria das categorias e modelando o sistema nervoso como um sistema complexo e emergente. Pesquisador do grupo Cognitio, o Núcleo de Apoia à Pesquisa em ciências cognitivas da Universidade de São Paulo. Participou recentemente de estágio de pesquisa no INRIA – Futurs, França.

João Eduardo Kögler Junior . Engenheiro Eletricista pelo Instituto Maua de Tecnologia e bacharel em Física pela Universidade de São Paulo , mestre em Engenharia Elétrica pela Universidade de São Paulo, com ênfase em engenharia biomédica e doutor em Engenharia Elétrica pela Universidade de São Paulo. Foi pesquisador visitante do Siemens Corporate Research, em Princeton, Estados Unidos, durante 1991-92 e do INRIA - Sophia Antipolis, na França, em 1994. Atualmente dedica-se à pesquisa e ensino na Escola Politécnica da Universidade de São Paulo, atuando nas áreas de Visão Computacional, Processamento de Imagens e Inteligência Computacional. É membro senior do IEEE – Computational Intelligence Society, da Sociedade Brasileira de Computação e da Sociedade Brasileira de Física. Além de ter atuado em pesquisa e ensino, também foi consultor e empresário na iniciativa privada, na área de Visão de Máquina e Automação Inteligente, como alliance member da National Instruments. Foi engenheiro de projetos e estudos de sistemas de geração e transmissão de energia elétrica da Companhia Energética de São Paulo e da Themag Engenharia.