

UNIVERSIDADE FEDERAL DE PELOTAS
Instituto de Física e Matemática
Departamento de Informática
Curso de Bacharelado em Ciência da Computação



Projeto de Conclusão de Curso

SINCRONIZAÇÃO DE AUTÔMATOS: UMA ABORDAGEM CATEGORIAL

ÉRICO KAZUHIRO SAWABE

PELOTAS, 2006

ÉRICO KAZUHIRO SAWABE

SINCRONIZAÇÃO DE AUTÔMATOS: UMA ABORDAGEM CATEGORIAL

Projeto de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Pelotas, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Carlos Antônio Pereira Campani, Dr.

PELOTAS, 2006

BANCA EXAMINADORA:

Prof. Carlos Antônio Pereira Campani, Dr. (Orientador)

Prof. Simone André da Costa Cavalheiro, MSc.

Prof. Sérgio Luiz Cardoso Oliveira, MSc.

AGRADECIMENTOS

Aos meus pais, Kandi Sawabe e Yumiko Kitamura Sawabe, por todo amor, carinho, amizade, apoio e por me mostrarem sempre que através da força de vontade, fé e muito trabalho, nós conseguimos atingir os objetivos que queremos.

Agradeço a Deus por estar sempre presente conduzindo da melhor forma a minha vida e que muitas vezes foi o único que conseguiu me escutar nos dias de tristeza, nervosismo, e angústia. Hoje, é mais um dia que sou grato a Ele.

À minha namorada Thaísa Leal por sempre estar me incentivando com sua alegria contagiante, disposição e por dividir comigo todos os momentos bons e ruins, sempre com uma palavra de otimismo.

A todos os professores que tive ao longo do curso, os quais me proporcionaram todo o conhecimento necessário para esta profissão. Agradeço em especial, aos professores José Luís Güntzel, Luciano Agostini, Gil Medeiros e Carlos Campani, verdadeiros exemplos para a UFPel, por tudo que aprendi durante minha formação acadêmica e pela amizade, respeito, compreensão, paciência e bons conselhos que eles souberam transmitir não só a mim, mas à todos os seus alunos.

Aos meus colegas com quem fiz amizade ao longo do curso, pelo companheirismo, os fantásticos jogos de futebol que foram organizados, as brincadeiras para descontrair, tornando as aulas mais divertidas, os estudos em grupo, as festas da turma e, principalmente, pelas grandes amizades conquistadas que foram muito além do convívio da universidade.

Ao professor Carlos Campani pela orientação no desenvolvimento deste trabalho, por todo o seu incentivo, otimismo e sua disposição, estando sempre pronto a me ajudar. Gostaria de agradecer também às pessoas que me ajudaram de uma maneira ou de outra, tanto no início do curso quanto nesta reta final, que nem me atrevo a colocar os nomes com receio de esquecer de algum deles. Meu sincero: Obrigado!!!

Resumo

SAWABE, Érico Kazuhiro. **Sincronização de Autômatos: Uma Abordagem Categórica**. 2006. 83f. Projeto de Conclusão de Curso (Graduação) – Curso de Bacharelado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Este trabalho aborda a sincronização de autômatos, tanto de autômatos não temporizados, quanto de autômatos temporizados, aplicando operações categoriais. A nova abordagem categorial da sincronização de autômatos temporizados, baseada na definição de autômatos temporizados proposta por Alur e Dill, visa modelar o comportamento de sistemas de tempo real. A importância de se tratar o sincronismo é que podemos expressar o comportamento de diversos tipos de sistemas em várias áreas que envolvem a computação.

A Teoria das Categorias fornece uma ferramenta matemática que pode manipular estruturas matemáticas complexas, cujas características notáveis são a capacidade de generalização, abstração, unificação e principalmente o fato do alto poder de expressividade de suas construções, permitindo formalizar idéias complexas de maneira simples, bem como propiciar um novo, ou melhor, entendimento das questões relacionadas a toda Ciência da Computação.

A nova abordagem categorial desenvolvida tem como objetivo tratar sincronização de autômatos, através da utilização do produto categorial e principalmente do produto fibrado baseado em grafos, permitindo gerar um autômato sincronizado com base em tabelas de sincronizações (restrições) adotadas, possibilitando ainda que uma parte do sistema original seja restringida e a outra mantida, preservando suas características.

A principal contribuição deste trabalho está na forma de se efetivar a sincronização de autômatos aplicada para autômatos temporizados, pela definição do modelo de Alur e Dill, utilizando a abordagem categorial com a operação produto. Ressaltando-se, contudo, que um autômato temporizado é um sistema de transição finito com restrições de relógio associadas a suas arestas e estados, que definem o comportamento dinâmico quanto ao tempo de permanência nos estados e os instantes disponíveis para transições. Além disso, é proporcionada uma outra forma de sincronização de autômatos temporizados utilizando a abordagem categorial com a operação de produto fibrado, aplicado ao resultado da composição de autômatos temporizados, determinando a sincronização. Esta operação possibilita que sistemas complexos possam ser definidos a partir do produto dos sistemas componentes.

O método de sincronização adotado visa sistematizar e expressar de forma compreensível a solução do sincronismo entre autômatos. Deste modo, algumas vantagens foram obtidas ao se aplicar as operações categoriais, como: não é necessário que se criem novos autômatos; os autômatos originais não são perdidos, podendo assim ser reutilizados quantas vezes forem necessários. A noção de reuso é uma das características marcantes em Teoria das Categorias.

Palavras-chave: Teoria das Categorias. Autômatos. Autômatos Temporizados. Sincronização de autômatos. Operações categoriais. Produto fibrado.

Abstract

SAWABE, Érico Kazuhiro. **Sincronização de Autômatos: Uma Abordagem Categorical**. 2006. 83f. Projeto de Conclusão de Curso (Graduação) – Curso de Bacharelado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

This work approaches the automata synchronization, comprising untimed automata as well as timed automata, applying categorical operations. The new categorical approach of timed automata synchronization, based on the timed automata definition proposed by Alur and Dill, aims to model real-time systems behavior. The importance to deal with the synchronism is that we can express several type of systems's behaviors in several Computer Science fields.

The Category Theory provides a mathematical tool which can manipulate complex mathematical structures. Its notable features are the ability of generalization, abstraction, unification and mainly the high expression ability of its constructions, allowing to formalize complex ideas in the simple way, and also to provide a new, or better, acknowledgment of the issues related to the whole Computer Science.

The new categorical approach developed has the aim to treat automata synchronization, through the categorical product use and mainly the pullback based in graphs, allowing to build a synchronized automata based in the synchronization's tables (restrictions) adopted. It's also allows a part of original's system to be restricted and another one to be kept, maintain its features.

The main contribution of this work is the way to perform the automata synchronization applied to timed automata, by the definition of Alur and Dill's model, using the categorical approach with the product operation. Although, it's important to highlight that timed automata is a finite transition system with clock's restrictions associated on its edges and states, which define the dynamic behavior as well as the time of permanence on states and also instants available for transitions. In addition, it is provided another way to timed automata synchronization using the categorical approach with the pullback operation, being applied to the result of the timed automata composition, establishing the synchronization. This operation allows that complex systems to be defined from the product of the component's systems.

The method of synchronization adopted aims at systemizing and expressing in an understandable way, the solution for the synchronism between automata. Thus, some advantages were obtained by applying categorical operations, such as: it is not necessary to create news automata; the original automata are not lost, they can be reused as many times it would be needed. The notion of reuse is one of the important features in Category Theory.

Keywords: Category Theory. Automata. Timed Automata. Automata synchronization. Categorical operations. Pullback.

LISTA DE ILUSTRAÇÕES

Figura 1 – Máquina composta por três partes	18
Figura 2 – Representação da função programa com um grafo	19
Figura 3 – Exemplo de diagrama de transição	20
Figura 4 – Exemplo da tabela de transição de estados.....	20
Figura 5 – Representação da função programa não-determinista como um grafo....	22
Figura 6 – Exemplo de um grafo da máquina de <i>Mealy</i>	24
Figura 7 – Exemplo de um grafo da máquina de <i>Moore</i>	24
Figura 8 – Atores especificados no modelo AGA	26
Figura 9 – Animação envolvendo dois atores AGA.....	29
Figura 10 – Produto de grafos reflexivos como combinador de sistemas	31
Figura 11 – Produto de sistemas e correspondente sincronização	31
Figura 12 – Exemplo de um autômato temporizado	34
Figura 13 – Ator <i>bicho</i> temporizado	37
Figura 14 – Autômato temporizado da cancela	40
Figura 15 – Autômato temporizado do controlador eletrônico	40
Figura 16 – Composição dos autômatos temporizados P (cancela) e C (controlador)	41
Figura 17 – Comportamento da composição $P//C$	42
Figura 18 – Homomorfismo de grafos preserva origem e destino.....	49
Figura 19 – Diagrama comutativo para produtos	50
Figura 20 – Diagrama comutativo para monomorfismo.....	52
Figura 21 – Diagrama comutativo para epimorfismo.....	52
Figura 22 – Diagrama de um isomorfismo	52
Figura 23 – Exemplo de equalizador em Set.....	54
Figura 24 – Exemplo de equalizador em Set com i sendo o sub-objeto maximal	54
Figura 25 – Diagrama comutativo para equalizador.....	55
Figura 26 – Diagrama comutativo para produto fibrado	56
Figura 27 – Produto fibrado como cálculo de uma restrição	57
Figura 28 – Restrição de uma função	57

Figura 29 – Exemplo de restrição de uma função	57
Figura 30 – Diagrama comutativo para produto de grafos	59
Figura 31 – Diagrama comutativo para produto fibrado de grafos	60
Figura 32 – Autômatos de uma animação.....	61
Figura 33 – Produto de grafos entre dois autômatos	62
Figura 34 – Resultado da aplicação das restrições ao produto.....	64
Figura 35 – Aplicação do produto fibrado baseado na restrição em Set.....	65
Figura 36 – Diagrama comutativo para produto de autômatos temporizados	69
Figura 37 – Sistemas de transição temporizados de uma animação	70
Figura 38 – Resultado do produto entre os sistemas com a restrição estabelecida..	71
Figura 39 – Sistemas de transição temporizados independentes	72
Figura 40 – Produto entre os dois sistemas de transição temporizados	73
Figura 41 – Resultado da restrição ao produto entre os autômatos temporizados ...	75
Figura 42 – Aplicação do produto fibrado na composição paralela dos sistemas componentes.....	76

LISTA DE TABELAS

Tabela 1 – Exemplos de categorias	45
---	----

LISTA DE ABREVIATURAS E SIGLAS

AF	Autômato Finito
AFD	Autômato Finito Determinístico
AFND	Autômato Finito Não Determinístico
AGA	Animações Gráficas baseadas na Teoria dos Autômatos
AGA-S	Animações Gráficas baseadas em Autômatos temporizados Sincronizados
Pullback	Produto Fibrado
Pullback Square	Quadrado do Produto Fibrado
Timed Automata	Autômatos Temporizados.

SUMÁRIO

RESUMO	5
ABSTRACT	6
LISTA DE ILUSTRAÇÕES	7
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS E SIGLAS	10
1 INTRODUÇÃO	12
1.1 ORGANIZAÇÃO DA MONOGRAFIA.....	14
2 AUTÔMATOS E SINCRONIZAÇÃO	15
2.1 CONCEITOS BÁSICOS	15
2.2 AUTÔMATO FINITO DETERMINÍSTICO	18
2.3 AUTÔMATO FINITO NÃO-DETERMINÍSTICO	21
2.4 AUTÔMATO FINITO COM SAÍDA.....	22
2.4.1 <i>Máquina de Mealy</i>	23
2.4.2 <i>Máquina de Moore</i>	24
2.5 SINCRONIZAÇÃO DE AUTÔMATOS	25
3 AUTÔMATOS TEMPORIZADOS	33
3.1 COMPOSIÇÃO	38
4 TEORIA DAS CATEGORIAS	43
4.1 CATEGORIA	44
4.2 DIAGRAMAS.....	46
4.3 CATEGORIA DOS GRAFOS.....	47
4.3.1 <i>Homomorfismo de grafos</i>	48
4.3.2 <i>Categoria Gr</i>	49
4.4 PRODUTO CATEGORIAL.....	50
4.5 MORFISMOS	51
4.6 SUB-OBJETO	52
4.7 EQUALIZADOR.....	53
4.8 PRODUTO FIBRADO.....	55
4.8.1 <i>Exemplo do Produto Fibrado como Restrição em Set</i>	56
5 ABORDAGEM CATEGORIAL DE SINCRONIZAÇÃO DE AUTÔMATOS	59
5.1 SINCRONIZAÇÃO DE AUTÔMATOS NÃO TEMPORIZADOS.....	59
5.2 SINCRONIZAÇÃO DE AUTÔMATOS TEMPORIZADOS	66
6 CONCLUSÕES E TRABALHOS FUTUROS	78
7 REFERÊNCIAS	81

1 Introdução

Estudos em áreas ainda pouco exploradas como a Teoria das Categorias vêm aumentando consideravelmente e motivando pesquisadores a descobrirem novas abordagens de estudo acerca de teorias até então desconhecidas, aplicando-as à Ciência da Computação. Esse é um ponto importante da abordagem desse trabalho, isto é, o estudo de Teoria das Categorias e posterior aplicação de operações categoriais sobre autômatos.

A Teoria das Categorias fornece um formalismo matemático que pode manipular estruturas matemáticas complexas proporcionando características notáveis como a capacidade de generalização, abstração e unificação. À primeira vista pode parecer estranho aplicar Teoria das Categorias em Ciência da Computação, afinal são duas áreas de atuação distintas. Apesar disso, possuem muito em comum e se enriquecem mutuamente.

Dentre as principais características da Teoria das Categorias que motivam a sua aplicação em Ciência da Computação destacam-se: independência de implementação, dualidade, herança de resultados e comparação de expressividade de formalismos, notação gráfica e, principalmente, o fato do alto poder de expressividade de suas construções, as quais, freqüentemente não possuem paralelo nas demais teorias (HAEUSLER; MENEZES, 2001). Essa expressividade permite formalizar idéias complexas de forma simples, bem como propicia um novo, ou melhor entendimento das questões relacionadas a toda Ciência da Computação (HAEUSLER; MENEZES, 2001; DIRETRIZES, 2005).

O uso de Teoria das Categorias tende a dar uma visão “geral” do problema a ser tratado, não se preocupando com os detalhes, pois ela trabalha com operações poderosas o suficiente para tratar o problema a ser resolvido em alto nível.

Este trabalho propõe uma nova abordagem de sincronização de autômatos baseada em operações categoriais.

A importância de se tratar a sincronização de autômatos é que podemos expressar através deste formalismo o comportamento de diversos tipos de sistemas em várias áreas que envolvem a computação, tais como: dados multimídia em banco

de dados, sistemas reativos, circuitos digitais em microeletrônica, controle de processos em sistemas operacionais, animações gráficas, modelagem de cursos (hiper-animações, hipermídia ou hipertexto) disponibilizados via *web*, modelos de concorrência como redes de petri, roteamento de transmissão de pacotes em redes de computadores, etc. Deste modo, a sincronização de autômatos proporciona a composição dos elementos que fazem parte de sistemas componentes determinando uma seqüência de eventos.

O método tradicional de sincronização de autômatos é baseado essencialmente em uma composição de sistemas de transição, que realiza a combinação de estados e transições, ou seja, do produto cartesiano dos autômatos (grafos rotulados e orientados para representação de sistemas), onde necessariamente é criado um novo autômato desta composição (HAEUSLER; MENEZES, 2001).

No entanto, o método de sincronização de autômatos que será desenvolvida nesse trabalho utilizará uma das noções mais poderosas de Teoria das Categorias: o *produto fibrado*, o qual tem por objetivo generalizar os equalizadores para morfismos com diferentes origens, permitindo gerar um autômato sincronizado com base em tabelas de sincronizações (restrições) adotadas (MENEZES, 1996), possibilitando ainda que uma parte do sistema original seja restringida e a outra mantida, preservando suas características. Esse método de sincronização foi adotado visando sistematizar e expressar de forma simplificada a solução do problema de sincronismo entre autômatos.

Além disso, esse trabalho proporciona o entendimento da sincronização de autômatos temporizados utilizando a abordagem categorial e apresenta, inclusive, como essa sincronização ocorrerá. Ressaltando-se, contudo, que um autômato temporizado é um sistema de transição finito com restrições de relógio associadas a suas arestas e estados (ALUR, 1999; ACCORSI; MENEZES; NEDEL, 2001). O produto fibrado, aplicado ao resultado da composição de autômatos temporizados, determina a sincronização. Esta operação possibilita que sistemas complexos possam ser definidos a partir do produto dos sistemas componentes (ACCORSI, 2002).

1.1 Organização da Monografia

O presente trabalho está organizado conforme a descrição a seguir.

O capítulo 2 apresenta alguns conceitos básicos sobre autômatos com o objetivo de ressaltar sua presença nas mais variadas situações de nosso cotidiano e facilitar a compreensão dos demais capítulos desta monografia.

O capítulo 3 apresenta o modelo de Autômatos Temporizados, o qual visa modelar o comportamento de sistemas de tempo real.

O capítulo 4, por sua vez, aborda o conceito de Teoria das Categorias e suas principais operações, as quais serão aplicadas no decorrer deste trabalho como uma poderosa ferramenta para o desenvolvimento do sincronismo entre os autômatos.

O capítulo 5 é dedicado a apresentar a abordagem categorial adotada nesse trabalho, a qual trata tanto da sincronização de autômatos não temporizados, através da utilização do produto categorial e do produto fibrado baseado em grafos, quanto da sincronização de autômatos temporizados, através das respectivas aplicações categoriais descritas anteriormente.

Finalmente, o capítulo 6 apresenta as conclusões e propostas para trabalhos futuros.

2 Autômatos e Sincronização

Em nosso dia-a-dia os autômatos estão presentes nas mais variadas situações. Um bom exemplo de um autômato finito é o mecanismo de controle de um elevador. Ele não lembra todas as requisições anteriores, mas somente o andar atual, a direção de movimento (para cima ou para baixo), e a relação de requisições pendentes. Outro exemplo são os semáforos, onde temos três estados: cor verde, amarela e vermelha, e mudanças de cor que levam à mudança dos estados.

Talvez a principal razão para estudar autômatos seja a diversidade de aplicações onde eles aparecem. É o funcionamento destes e a compreensão da sua utilidade que pretende-se transmitir (AUTÔMATOS, 2006).

2.1 Conceitos Básicos

Uma linguagem é um meio de comunicação, formada por um conjunto de palavras e de regras gramaticais que permitem combinar as palavras em sentenças sintaticamente corretas. A linguagem é dita formal quando pode ser representada através de um sistema com sustentação matemática que compreende a representação da sintaxe (estrutura) e da semântica (significado) das sentenças de uma linguagem. Logo abaixo será abordada somente a estrutura sintática das linguagens (MENEZES, 1997; HOPCROFT; MOTWANI; ULLMAN, 2001; REIS, 2002; CAMPANI, 2006).

Definição 1 *Alfabeto (ou vocabulário) é um conjunto finito de símbolos (ou caracteres).*

Definição 2 *Sentença (ou palavra, ou cadeia de caracteres) a sentença sobre um alfabeto é uma seqüência finita de símbolos concatenados.*

Definição 3 *Linguagem é qualquer conjunto de sentenças sobre um alfabeto. Conforme o número de sentenças que possuem, as linguagens se classificam em finitas, infinitas e vazias.*

Gramática serve para definir qual o subconjunto de sentenças que faz parte de uma determinada linguagem. É um dispositivo formal para especificar uma linguagem potencialmente infinita de forma finita.

Definição 4 Formalmente uma gramática G é definida por uma quatro-upla:

$$G = (V, T, P, S)$$

Onde:

V – conjunto finito de símbolos não terminais ou variáveis (convencionalmente são adotadas letras maiúsculas);

T – conjunto finito de símbolos terminais (convencionalmente são adotadas letras minúsculas);

P – conjunto finito de pares (α, β) é representado por $\alpha \rightarrow \beta$, denominados regras de produção - tal que a primeira componente é uma palavra de $(V \cup T)^+$ e a segunda componente é a palavra de $(V \cup T)^*$ - que definem as condições de geração das palavras da linguagem.

$$P = \{\alpha \rightarrow \beta \mid \alpha \in N^+ \wedge \beta \in N^*\} \text{ em que } N = V \cup T;$$

S – elemento de V denominado variável inicial da gramática.

Uma seqüência de regras de produções da forma $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$ (mesma componente do lado esquerdo) pode ser abreviada como uma única produção na forma:

$$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

A aplicação de uma regra de produção é denominada *derivação* de uma palavra. A aplicação sucessiva de regras de produção permite derivar as palavras da linguagem representada pela gramática.

Definição 5 Derivação (\Rightarrow) é uma operação de substituição efetuada de acordo com as regras de produção da gramática $G = (V, T, P, S)$. É constituída por um par da relação denotada por \Rightarrow com domínio $(V \cup T)^+$ e contra-domínio em $(V \cup T)^*$. Um par (α, β) é representado de forma infixada, como segue: $\alpha \Rightarrow \beta$. Assim:

- Para toda a produção da forma $S \rightarrow \beta$ (primeira componente é o símbolo inicial de G) tem-se que: $S \Rightarrow \beta$.

- E para todo o par $\alpha \Rightarrow \beta$, onde $\beta = \beta_u \beta_v \beta_w$, se $\beta_v \rightarrow \beta_t$ é uma regra de P então: $\alpha \Rightarrow \beta_u \beta_t \beta_w$.

$$\text{Logo: } \gamma \alpha \delta \Rightarrow \gamma \beta \delta, \text{ se } \alpha \rightarrow \beta \in P \text{ e } \gamma, \delta \in V^*.$$

Definição 6 Linguagem gerada por uma gramática $G = (V, T, P, S)$, denotada por $L(G)$ ou $Gera(G)$, é o conjunto de todas as sentenças que podem ser geradas a partir do símbolo inicial S , através de derivações sucessivas, ou seja:

$$L(G) = \{w \mid w \in T^* \wedge S \Rightarrow^* w\}$$

Exemplo 1 Logo a seguir são fornecidos um exemplo de cada uma das definições anteriores.

Alfabeto: letras ($\{a,b,c,\dots,z\}$), dígitos ($\{0,1\}$), etc.

Sentença: Seja o alfabeto $V = \{0,1\}$, são sentenças válidas 010, 1101, etc.

Linguagem: Suponha o alfabeto $\Sigma = \{a,b\}$; o conjunto de palíndromos (palavras que tem a mesma leitura da esquerda para a direita e vice-versa) é um exemplo de uma linguagem infinita. Assim, $a, b, aba, bb, bab, aaaa, \dots$ são palavras desta linguagem.

Gramática: Dada uma gramática $G = (V, T, P, I)$ temos a representação de suas componentes logo abaixo:

$$V = \{I, B\}$$

$$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$P = (I \rightarrow IB, B \rightarrow 0, B \rightarrow 1, B \rightarrow 2, B \rightarrow 3, B \rightarrow 4, B \rightarrow 5, B \rightarrow 6, B \rightarrow 7, B \rightarrow 8, B \rightarrow 9) \text{ ou } P = I \rightarrow IB \mid B \rightarrow 0|1|2|3|4|5|6|7|8|9.$$

Derivação: Dada uma gramática $G = (\{S, A, B\}, \{0, 1\}, \{S \rightarrow AB, A \rightarrow 0|AB, B \rightarrow 1|1B\}, S)$, uma derivação do número 0111 é:

$$S \Rightarrow AB \Rightarrow ABB \Rightarrow 0BB \Rightarrow 01B \Rightarrow 011B \Rightarrow 0111.$$

Linguagem Gerada: Seja uma gramática $G = (\{S\}, \{0,1\}, P, S)$ com $P = \{S \rightarrow 0S1, S \rightarrow 01\}$. Então: $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0^3S1^3 \Rightarrow \dots 0^nS1^n \Rightarrow$ que é igual a 0^n1^n para $n \geq 1$.

$$\text{Logo, } L(G) = \{0^n1^n \mid n \geq 1\}.$$

Um autômato é definido como sendo um modelo matemático de uma máquina de estados finitos. Ele funciona como um reconhecedor de uma determinada linguagem e serve para modelar uma máquina. Um conceito fundamental nos autômatos é o conceito de estado. Este conceito é aplicado a qualquer sistema (AUTÔMATO, 2006).

Os autômatos finitos constituem um modelo útil para muitos elementos importantes de hardware e software. Alguns exemplos:

- Software para projetar e verificar comportamento de circuitos digitais.
- Analisadores léxicos de um compilador típico (isto é, componente que divide o texto de entrada em unidades lógicas, como identificadores, palavras-chave, etc.) e processadores de texto.
- Software para examinar grandes corpos de texto, como páginas Web.
- Software para verificar sistemas de todos os tipos que têm um número finito de estados distintos, como protocolos de comunicação ou segurança.

2.2 Autômato Finito Determinístico

Um autômato finito (AF) ou autômato finito determinístico (AFD) pode ser visto como uma máquina composta de três partes (MENEZES, 1997), conforme a Fig. 1:

- *Fita*: Dispositivo de entrada que contém a informação a ser processada;
- *Unidade de Controle*: Reflete o estado corrente da máquina. Possui uma unidade de leitura que lê o símbolo de uma célula de cada vez e movimenta-se exclusivamente para a direita;
- *Função de Transição ou Programa*: Função que comanda as leituras e define o estado da máquina;



Figura 1 – Máquina composta por três partes

Um autômato finito é representado através de um controle finito, que tem acesso a uma fita onde está a seqüência a ser analisada. O autômato percorre esta fita da esquerda para a direita, lendo um símbolo de cada vez. Estando o controle em um estado, apontando para um determinado símbolo na entrada, ocorre uma

transição de estado, que faz com que o controle passe para outro estado e avance para o próximo símbolo na entrada. Se isto se repetir com sucesso até o final da fita de entrada e, no final, o autômato estiver em um estado final, a seqüência foi reconhecida ou aceita. Se, ao contrário, ocorrer alguma falha (transição impossível) o reconhecimento pára e a seqüência não é reconhecida, ou seja, ela é rejeitada.

Definição 7 *Formalmente, um autômato finito M é uma cinco-upla:*

$$M = (\Sigma, Q, \delta, q_0, F)$$

Onde:

Σ – alfabeto de símbolos de entrada do autômato e corresponde a um conjunto finito não vazio dos símbolos de entrada ou átomos indivisíveis que compõem a cadeia de entrada submetida ao autômato para aceitação;

Q – conjunto finito, não vazio, de estados do autômato finito;

δ – função programa ou função de transição de estados do autômato é indicar as transições possíveis em cada configuração do autômato.

$$\delta: Q \times \Sigma \rightarrow Q$$

Esta função fornece para cada par (estado, símbolo de entrada) um novo estado para onde o autômato deverá mover-se. E pode ser representada como um grafo finito direto, como ilustrado na Fig. 2.

q_0 – estado inicial do autômato finito, e corresponde a um elemento do conjunto Q ;

F – conjunto de estados finais tal que F está contido em Q .

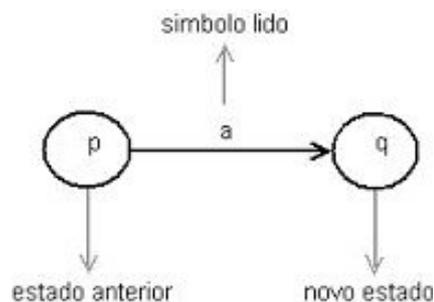


Figura 2 – Representação da função programa com um grafo

É um autômato que se encontra sempre em um único estado após ler uma seqüência qualquer de entrada. O termo “*determinístico*” implica que existe um e somente um estado ao qual o autômato pode transitar a partir de seu estado atual. Em

contraste, autômatos finitos *não-determinísticos* podem ter várias alternativas de estados válidos para transição (MENEZES, 1997; REIS, 2002).

Uma outra maneira de representar um autômato finito são os *diagramas de transições* (REIS, 2002). Trata-se de um grafo direcionado e rotulado. Os vértices representam os estados, sendo representados por círculos. Os estados finais são representados por dois círculos concêntricos e o estado inicial é indicado por uma seta. As arestas representam as transições entre dois estados, sendo o rótulo o símbolo reconhecido. A Fig. 3 apresenta um exemplo de um diagrama de transição.

Exemplo 2 Dado um autômato $M = (\{a,b\}, \{q_0,q_1,q_2\}, \delta, q_0, \{q_2\})$, e $\delta(q_0, a) = q_1$, $\delta(q_1, a) = q_1$, $\delta(q_1, b) = q_2$. Temos:

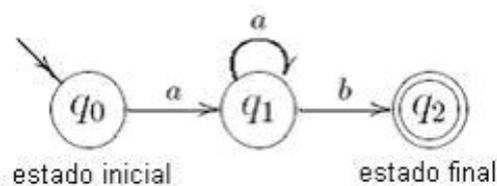


Figura 3 – Exemplo de diagrama de transição

Logo, $T(M) = \{a^n b \mid n \geq 1\}$, onde T é uma linguagem gerada pela gramática M .

Uma terceira maneira de representar um autômato finito é através de uma *tabela de transição* (HOPCROFT, MOTWANI, ULLMAN, 2001; REIS, 2002; CAMPANI, 2006), indicando na vertical os estados e na horizontal os símbolos do alfabeto. O estado inicial é indicado com uma seta e os finais com asteriscos. O conteúdo da tabela indica as transições de estado possíveis. A Fig. 4 apresenta um exemplo de uma tabela de transição de estados de acordo com o diagrama do exemplo anterior.

δ	a	b
$\rightarrow q_0$	q_1	-
q_1	q_1	q_2
$*q_2$	-	-

Figura 4 – Exemplo da tabela de transição de estados

2.3 Autômato Finito Não-Determinístico

Um autômato finito não-determinístico (AFND) tem a possibilidade de mais de uma transição de estados para cada estado. Essa habilidade é expressa com freqüência como a capacidade de “adivinhar” algo sobre sua entrada. Por exemplo, quando o autômato é usado para procurar certas seqüências de caracteres (como uma palavra-chave) em uma longa *string* de texto, é útil “adivinhar” que estamos no início de uma dessas *strings* e usar uma seqüência de estados apenas para verificar se a *string* aparece, caractere por caractere.

Os AFND aceitam exatamente as linguagens regulares, da mesma maneira que fazem os AFDs. Desta forma sempre é possível converter um AFND em um AFD, porém o AFD gerado pode ter exponencialmente mais estados que o AFND. Muitas vezes os AFNDs são mais sucintos e mais fáceis de projetar que um AFD (REIS, 2002).

A diferença entre um AFD e um AFND está na função de transição δ . Para um AFND esta função recebe um estado e um símbolo de entrada como argumentos (da mesma forma que um AFD), porém retorna um conjunto de zero, um ou mais estados (em vez de retornar exatamente um estado como um AFD deve fazer).

Definição 8 *Formalmente, um autômato finito não-determinístico é uma cinco-upla (HOPCROFT, MOTWANI, ULLMAN, 2001; CAMPANI, 2006):*

$$M = (\Sigma, Q, \delta, q_0, F)$$

Onde:

Σ – alfabeto finito de símbolos de entrada;

Q – conjunto finito, não vazio, de estados possíveis do autômato;

δ – função de transição de estados, mapeando $Q \times \Sigma \rightarrow \text{Partes}(Q)$ ou $Q \times \Sigma \rightarrow 2^Q$ (é o conjunto de todos os subconjuntos de Q). Isto é, $\delta(q, a)$ é um subconjunto de Q para cada estado q e símbolo de entrada a . Assim, $\delta(q, a) = \{p_1, p_2, \dots, p_k\}$ significa que M estando no estado q e tendo a na posição do cabeçote na fita de entrada, move uma célula para a direita e escolhe qualquer estado entre p_1, p_2, \dots, p_k como próximo estado. A função programa pode ser representada como um grafo finito direto, conforme a Fig. 5;

q_0 – estado inicial e $q_0 \in Q$;

F – conjunto de estados finais e $F \in Q$.

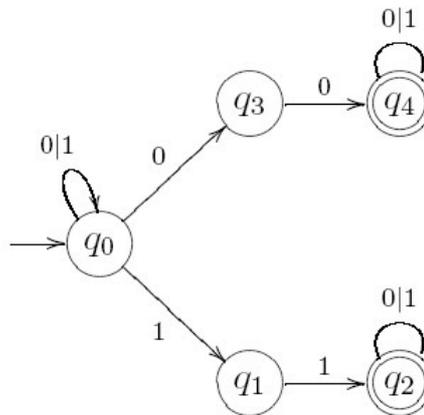


Figura 5 – Representação da função programa não-determinista como um grafo.

Exemplo 3 Dado um autômato não-determinístico $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_2, q_3, q_4\})$, temos os seguintes resultados da função de transição de estados, conforme representado na Fig. 5:

$$\delta(q_0, 0) = \{q_0, q_3\}$$

$$\delta(q_2, 1) = \{q_2\}$$

$$\delta(q_0, 1) = \{q_0, q_1\}$$

$$\delta(q_3, 0) = \{q_4\}$$

$$\delta(q_1, 0) = \emptyset$$

$$\delta(q_3, 1) = \emptyset$$

$$\delta(q_1, 1) = \{q_2\}$$

$$\delta(q_4, 0) = \{q_4\}$$

$$\delta(q_2, 0) = \{q_2\}$$

$$\delta(q_4, 1) = \{q_4\}$$

2.4 Autômato Finito com Saída

O conceito básico de autômatos finitos possui aplicações restritas, pois a informação de saída é limitada à lógica binária aceita/rejeita. Sem alterar a classe de linguagens reconhecidas, é possível estender a definição de autômato finito, incluindo a geração de uma palavra de saída, denominando assim um Autômato Finito com Saída (MENEZES, 1997; BRITO; MARTENDAL; OLIVEIRA, 2006). As saídas podem ser associadas às transições (Máquina de *Mealy*) ou aos estados (Máquina de *Moore*). Em ambos os casos a saída não pode ser lida, ou seja, não pode ser usada como memória auxiliar, e suas características são:

- é definida sobre um alfabeto especial, denominado alfabeto de saída (pode ser igual ao alfabeto de entrada);

- a saída é armazenada em uma fita independente da fita de entrada;
- a cabeça da fita de saída move uma célula para direita a cada símbolo gravado;
- o resultado do processamento do autômato finito é o seu estado final (condição de aceita/rejeita) e a informação contida na fita de saída.

As máquinas de *Mealy* e *Moore* abordadas são modificações sobre o autômato finito determinístico e serão explicadas a seguir (MENEZES, 1997; CAMPANI, 2006; BRITO; MARTENDAL; OLIVEIRA, 2006).

2.4.1 Máquina de *Mealy*

Definição 9 *Uma máquina de Mealy é autômato finito determinístico modificado de forma a gerar uma palavra de saída para cada transição entre os estados. Neste tipo de máquina de estados às palavras de saída dependem do estado atual e do valor das entradas.*

Tal máquina pode ser representada formalmente pela seis-upla:

$$ME = (\Sigma, Q, \delta, q_0, F, \Delta)$$

Onde:

Σ – alfabeto de símbolos de entrada;

Q – conjunto de estados possíveis do autômato, o qual é finito;

δ – função programa ou de transição $\delta: Q \times \Sigma \rightarrow Q \times \Delta^*$;

q_0 – estado inicial do autômato, tal que q_0 é elemento de Q ;

F – conjunto de estados finais, tal que F está contido em Q ;

Δ – alfabeto de símbolos de saída.

O processamento de uma máquina de *Mealy* para uma dada entrada w consiste na aplicação sucessiva da função programa para cada símbolo de w (da esquerda para a direita), até ocorrer uma condição de parada. Caso a saída da função programa seja uma palavra vazia, nenhuma gravação é realizada, ou seja, a cabeça da fita não se move. Porém se todas as transições de uma determinada máquina de *Mealy* gerarem saídas vazias, então esta se comporta como um autômato finito. A Fig. 6 abaixo é um exemplo de um grafo da máquina de *Mealy*.

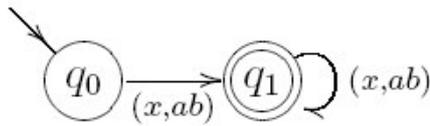


Figura 6 – Exemplo de um grafo da máquina de *Mealy*

As arestas são rotuladas com um par (x, y) , onde x é o símbolo lido e y é a string produzida na saída.

2.4.2 Máquina de *Moore*

Definição 10 Uma máquina de Moore é um autômato finito determinístico com uma função que gera uma palavra de saída (que pode ser vazia) para cada estado da máquina. Esta saída só depende de seu estado atual.

Tal máquina é representada formalmente por uma sete-upla:

$$MO = (\Sigma, Q, \delta, q_0, F, \Delta, \delta S)$$

Onde:

Σ – alfabeto de símbolos de entrada;

Q – conjunto de estados possíveis do autômato, o qual é finito;

δ – função programa ou de transição $\delta: Q \times \Sigma \rightarrow Q$;

q_0 – estado inicial do autômato, tal que q_0 é elemento de Q ;

F – conjunto de estados finais tal que F está contido em Q ;

Δ – alfabeto de símbolos de saída;

δS – função de saída $\delta S: Q \rightarrow \Delta^*$ a qual é uma função total.

O processamento de uma máquina de *Moore* ocorre de forma semelhante que na máquina de *Mealy*, assim como o tratamento de saídas vazias. Do mesmo modo que a máquina de *Mealy*, se todos os seus estados gerarem saída vazia, a máquina de *Moore* também se comportará como um autômato finito. A Fig. 7 abaixo é um exemplo de um grafo da máquina de *Moore*.

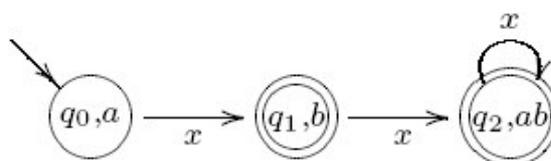


Figura 7 – Exemplo de um grafo da máquina de *Moore*

Os vértices são rotulados com um par (x, y) , onde x é o nome do estado e y é a saída produzida pelo estado.

Os dois modelos de autômato finito com saída são capazes de produzir o mesmo mapeamento da entrada para a saída, ou seja, há equivalência entre eles. Contudo, esta equivalência não é válida para a entrada vazia. Essa exceção não é relevante nesta monografia, posto que o modelo de sincronização de autômatos adotado na mesma, desconsidera tal situação como entrada.

2.5 Sincronização de autômatos

Vários estudos baseados em autômatos finitos vêm sendo efetuados. O trabalho de Machado (MACHADO, 2000) foi um dos pioneiros na utilização dos autômatos para *Hyper-Documentos na Web*, tratando-se de um ambiente semi-automatizado para o suporte ao gerenciamento de hipertextos destinados ao ensino à distância, cuja arquitetura está baseada em um modelo de organização de hiperdocumentos definido sobre o formalismo de autômatos finitos com saída. Saito (SAITO, 2001) estuda uma nova forma de visualizar os autômatos de uma maneira mais clara e menos complexa, pretendendo com isto automatizar a tarefa de criação do autômato. Accorsi (ACCORSI; MENEZES, 2000; ACCORSI, 2002) apresenta o modelo de Animações Gráficas baseadas na Teoria dos Autômatos (AGA), criando animações computacionais baseadas em autômatos finitos, já com alguns protótipos implementados. Este modelo será mais detalhado a seguir, pois sua utilização servirá como referência para melhor definir a sincronização de autômatos.

O modelo AGA, de maneira semelhante às animações em tempo real, especifica a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação (HOPCROFT; ULLMAN, 1979). Tal modelo define as alterações dos atores na animação a partir de uma estrutura baseada em autômatos finitos com saída (MENEZES, 1997).

Cada um dos atores na animação é especificado a partir de uma extensão para o autômato com saída, a qual vincula imagens do ator à saída do autômato. Deste modo, quando os autômatos são simulados, mediante a leitura da fita de entrada, as transições entre seus estados controlam a animação dos atores.

A área de atuação dos atores é dividida em camadas com o objetivo de estabelecer a ordem de sobreposição das imagens. Na Fig. 8, a imagem do ator *maçã* (camada 1) deve ser sobreposta pela imagem do ator *bicho* (camada 2) quando ocuparem o mesmo lugar. Assim, a imagem correspondente a cada instante da animação é formada pela unificação das camadas. A cada transição é associada uma imagem, de forma que estando, por exemplo, no estado $q1$ e o símbolo 2 for lido da fita de entrada, o ator *bicho* irá exibir a imagem da minhoca contraída e irá para o estado $q4$.

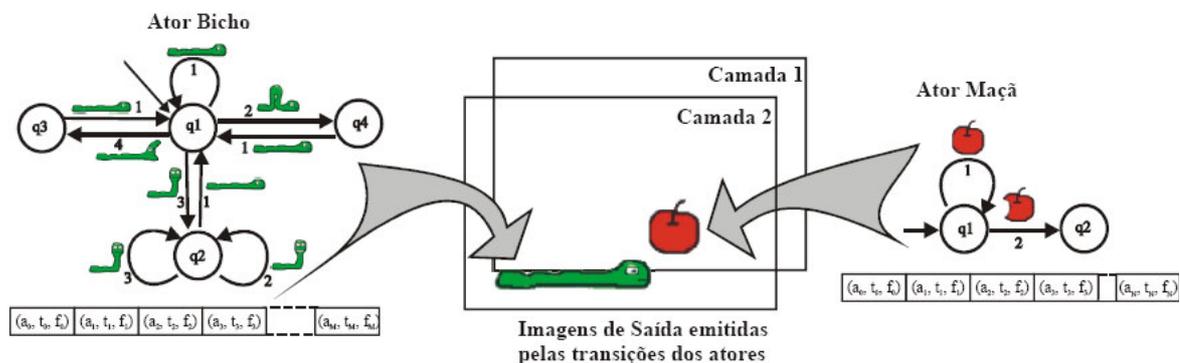


Figura 8 – Atores especificados no modelo AGA

A fita de entrada utilizada pelo autômato deverá possuir três elementos. Sendo que o primeiro elemento é o símbolo de entrada, o segundo corresponde ao instante em que devem ocorrer as transições e o terceiro elemento é o conjunto de transformações aplicado à imagem associada à saída. Esse conjunto de transformações, também denominado de *funções de controle*, é responsável por diferentes tarefas, como por exemplo, posicionar o ator na área de animação.

A estrutura do modelo AGA é baseada em uma unidade básica, chamada de *ator AGA*, a qual é definida a partir de extensões propostas para os autômatos com saída. Essas extensões complementam a definição de autômato com saída com o objetivo de: suportar o controle temporal das transições; prover imagens como saída com adaptações contextuais e, associar descrições semânticas aos estados.

Devido à equivalência entre as duas abordagens de autômatos com saída, o ator AGA pode ser estendido tanto vinculando a saída às transições quanto aos estados.

Contudo, a extensão do ator AGA a partir da máquina de *Mealy* foi preferida para a apresentação e implementação do modelo, pois como associa as imagens de

saída às transições, os estados podem ser incluídos de acordo com a modelagem desejada para o ator, independente do número de imagens no alfabeto de saída.

Definição 11 *O ator AGA é definido então como uma nove-upla:*

$$At = (Q, \Sigma, \Delta, \delta_t, \lambda_c, q_0, \sigma, F, D)$$

Onde:

Q – conjunto finito de estados possíveis do autômato;

Σ – alfabeto de símbolos de entrada;

Δ – alfabeto de símbolos de saída;

δ_t – função de transição temporal (função parcial $\delta_t: Q \times \Sigma \times \mathbb{N} \rightarrow Q$) para a qual, a especificação $\delta_t(q, a, t) = q'$ determina que deve ocorrer uma transição de q para q' quando o símbolo a é lido, e depois, aguardado o tempo t no estado q' antes da leitura do próximo símbolo;

λ_c – função de saída contextual (função parcial $\lambda_c: Q \times \Sigma \times F^* \rightarrow \Delta^F$), onde F^* é o conjunto de cadeias $\varphi = f_1 f_2 \dots f_n$ formadas por zero ou mais funções de F . Logo, $\lambda_c(q, a, \varphi)$ determina que uma imagem de saída está associada à transição a que diverge de q . Uma cadeia φ é associada a cada um dos símbolos da fita de entrada para determinar quais transformações devem ser aplicadas a cada transição do autômato.

q_0 – estado inicial do autômato ($q_0 \in Q$);

σ – função descrição (função parcial $\sigma: Q \rightarrow D$) define D é o conjunto de descrições utilizadas pelo ator AGA, mas nem todo estado precisa necessariamente estar mapeado para alguma descrição em D ;

F – conjunto finito de funções de transformação, cujos elementos são funções do tipo $f(v^{\rightarrow}, i) = i^{\dagger}$, onde i e i^{\dagger} são elementos de Δ^F e $v^{\rightarrow} = v_1 v_2 \dots v_n$ é um vetor de argumentos necessários para o mapeamento de i para i^{\dagger} . O conjunto Δ^F é composto por todas as imagens que podem ser produzidas pela composição de zero ou mais imagens de Δ transformadas por zero ou mais funções, ou seja, deste modo dependendo do instante da animação, a imagem de saída pode sofrer transformações com o objetivo de se adequar ao contexto da cena representada.

D - conjunto finito de descrições.

A fita de entrada utilizada pelo ator AGA é limitada ao tamanho da entrada e definida como $ft = (a_1, t_1, \varphi_1)(a_2, t_2, \varphi_2) \dots (a_n, t_n, \varphi_n)$, na qual cada célula possui a tripla

ordenada $(a, t, \varphi) \in \Sigma \times N \times F^*$, onde a é um símbolo do alfabeto de entrada, t é um tempo em milésimos de segundo e φ uma cadeia de funções de transformação de saída.

Portanto, seja A um conjunto de atores AGA e T um conjunto de fitas de entrada para os atores AGA, uma animação no modelo AGA, ou simplesmente AGA, é um conjunto totalmente ordenado de pares ordenados $(At, ft) \in A \times T$, cuja relação de ordem total, denotada por \leq , é definida como: $(At_1, ft_1) \leq (At_2, ft_2)$ se e somente se At_1 atua em uma camada menor ou igual à camada de At_2 na área de animação. Assim, $AGA = (\{(At, ft) \mid At \in A \text{ e } ft \in T\}, \leq)$, onde todos os atores iniciam o processamento de suas fitas de entrada no mesmo instante.

Desta maneira, a apresentação de uma animação modelada em AGA pode ser obtida a partir da simulação dos atores AGA, na qual estes processam em paralelo suas fitas de entrada correspondentes, e emitem as imagens de saída em uma mesma cena animação, respeitando as camadas determinadas pela relação de ordem.

Como exemplo tem-se a Fig. 9 que ilustra a modelagem em AGA. A animação tem duração de 1900ms e cada uma das imagens representa a situação dos atores a partir do instante indicado no canto superior esquerdo.

A modelagem da animação é criada a partir de dois atores: *Bicho* e *Maçã*, os quais são especificados como segue:

- $Bicho = (\{q_1, q_2, q_3, q_4\}, \{1, 2, 3, 4\}, \Delta_1, \delta_{t1}, \lambda_{c1}, q_1, \sigma_1, \{Trans(x,y)\}, \{“Boca Aberta”, “Contraído”, “Alerta”\})$, onde Δ_1, δ_{t1} e λ_{c1} estão descritos na Fig. 8 (Ator *Bicho*). A imagem mais próxima à transição indica a associação da saída. A função σ_1 é mapeada como: $\sigma_1(q_2) = “Alerta”, \sigma_1(q_3) = “Boca Aberta”$ e $\sigma_1(q_4) = “Contraída”$.

- $Maçã = (\{q_1, q_2\}, \{1, 2\}, \Delta_2, \delta_{t2}, \lambda_{c2}, q_1, \sigma_2, \{Trans(x,y)\}, \{“Mordida”\})$, onde Δ_2, δ_{t2} e λ_{c2} estão descritos na Fig. 8 (Ator *Maçã*) e $\sigma_2(q_2) = “Mordida”$.

A função $Trans(x,y)$, pertencente ao conjunto de funções F de cada ator, efetua a translação da imagem de saída nos eixos X e Y . A origem dos eixos é o canto superior esquerdo da área de animação.

Para esta animação são especificadas duas fitas de entrada, uma para o ator *Bicho* e outra para o ator *Maçã*, respectivamente, como segue:

- $ft_{Bicho} = (1, 200, Trans(2,50))(2, 200, Trans(32,0))(1, 200, Trans(32,0))(4, 200, Trans(32,0))(1, 200, Trans(32,0))(3, 200, Trans(32,0))(2, 500, Trans(32,0))(3, 200, Trans(32,0))$.
- $ft_{Maçã} = (1, 800, Trans(142,70))(2, 1100, Trans(142, 70))$.

Portanto, a animação é especificada como $AGA = ((Bicho, ft_{Bicho}), (Maçã, ft_{Maçã}))$, onde $Maçã < Bicho$. Note que o modelo descreve todo o comportamento dos atores na animação de tal forma que a simulação dos autômatos pode ser utilizada para compor as imagens de saída em qualquer um dos instantes do intervalo 0 até 1900ms. Na Fig. 9, abaixo de cada imagem, são representados os estados correntes de cada ator e suas transições durante a simulação. Note que, primeiramente, ambos os atores ocupam o estado inicial, e no instante 0 com a leitura da primeira célula de cada fita, as transições dos atores são provocadas emitindo as respectivas imagens de saída.

Os atores realizam suas transições, indicadas pelas setas na Fig. 9, de forma independente mediante a leitura do conteúdo de cada fita. Na animação representada, o ator *Maçã* realiza apenas duas transições durante a animação, enquanto o outro ator realiza oito. As fitas foram construídas para que o fechamento da boca do bicho coincida com a troca de imagem da maçã.

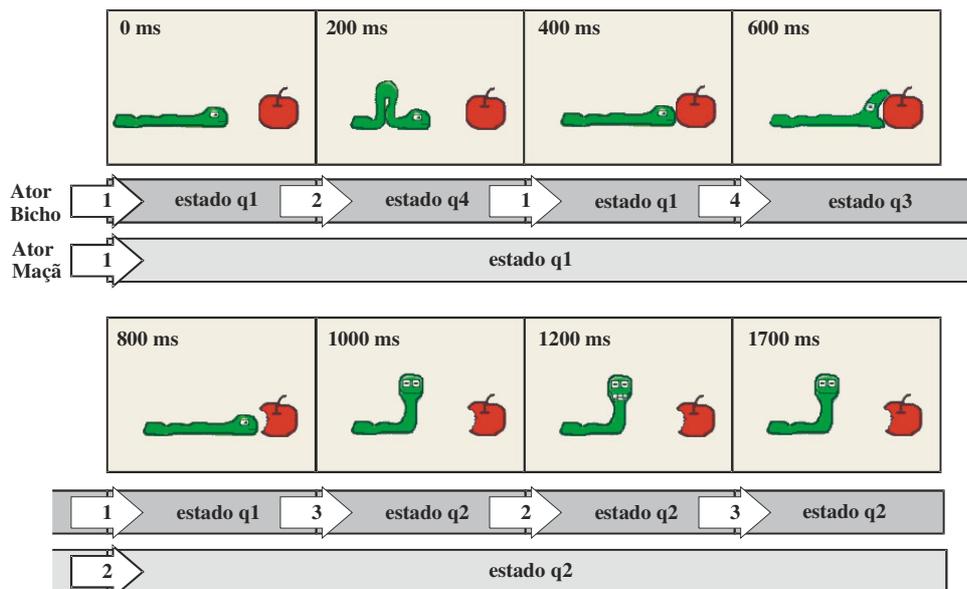


Figura 9 – Animação envolvendo dois atores AGA

No modelo AGA não é fundamentado matematicamente, como se podem criar novas animações a partir dos atores originais. O que se subentende é que será necessário que se criem novos autômatos ou que se modifiquem as fitas de entrada de todos os atores envolvidos na animação.

Contudo, esse exemplo é bastante relevante, pois apresenta de forma clara e didática, através da divisão em camadas, com a respectiva sobreposição de imagens, o que ocorre no processo de sincronização de autômatos.

A sincronização de autômatos pode também, ser realizada através do produto cartesiano – conjunto de todas as combinações possíveis das transições entre os estados – entre os autômatos (grafos rotulados e orientados para representação de sistemas) que se deseja sincronizar e, a partir desse produto cartesiano, suprimir os estados e as transições que não serão utilizados, restando somente o conjunto de estados e transições pertinentes à sincronização almejada (HAEUSLER; MENEZES, 2001).

Um exemplo desse processo de sincronismo é mostrado a partir da Fig. 10, onde considerando os grafos reflexivos (esquerda e centro) os quais representam os sistemas *Flip-Flop* e *Relógio*. Uma transição identidade de um nodo é interpretada como uma transição do tipo *sem operação*, a qual não altera o estado do sistema.

O grafo resultante do produto cartesiano ilustrado na Fig. 10 (direita) é um sistema no qual as transições representam todas as combinações *síncronas e assíncronas* de *flip* e *flop* com *tic* e *tac*.

Notamos que, por exemplo, a transição $\langle flip, t_x \rangle: \langle A, X \rangle \rightarrow \langle B, X \rangle$, etiquetada simplesmente por *flip*, é a ocorrência de *flip* juntamente com *sem operação* e, portanto, pode ser interpretada como simples ocorrência de *flip*. Por simplicidade, as transições: $\langle flip, tic \rangle$, $\langle flip, tac \rangle$, $\langle flop, tic \rangle$ e $\langle flop, tac \rangle$ são abreviadas por $\langle fi, ti \rangle$, $\langle fi, ta \rangle$, $\langle fo, ti \rangle$ e $\langle fo, ta \rangle$, respectivamente.

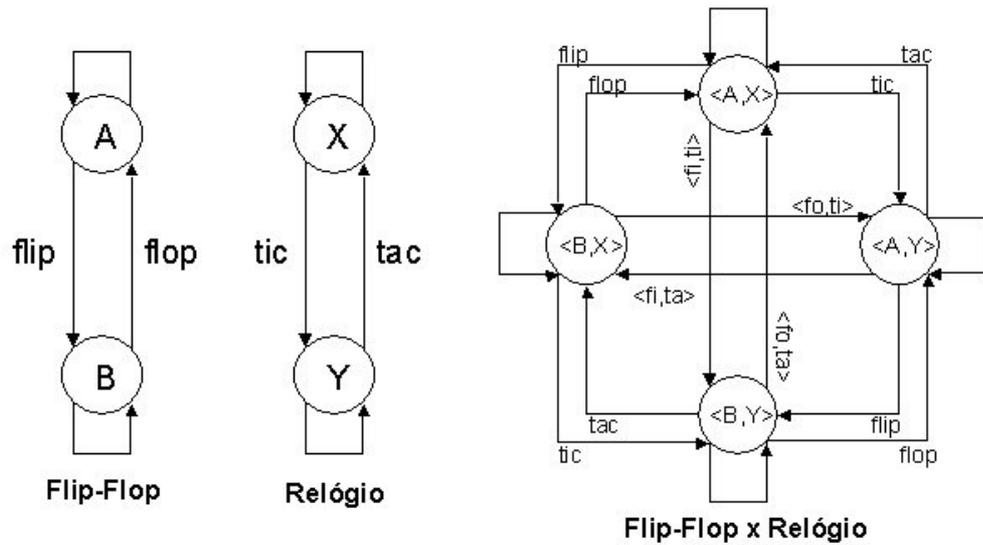


Figura 10 – Produto de grafos reflexivos como combinador de sistemas

Caso, no sistema *Flip-Flop x Relógio* da Fig. 10, for desejado, por exemplo, sincronizar as transições de *flip* com *tic* e permitir que *flop* e *tac* ocorram livremente, ou seja, que somente as seguintes transições ocorram: $\langle flip, tic \rangle$, $flop, tac$ e $\langle flop, tac \rangle$.

Então, é suficiente “restringir” o sistema resultante, ou seja, “apagar” as transições etiquetadas por *flip* e *tic*, como ilustrado na Fig. 11 (direita), no qual as transições “apagadas” estão representadas com um traço diferenciado.

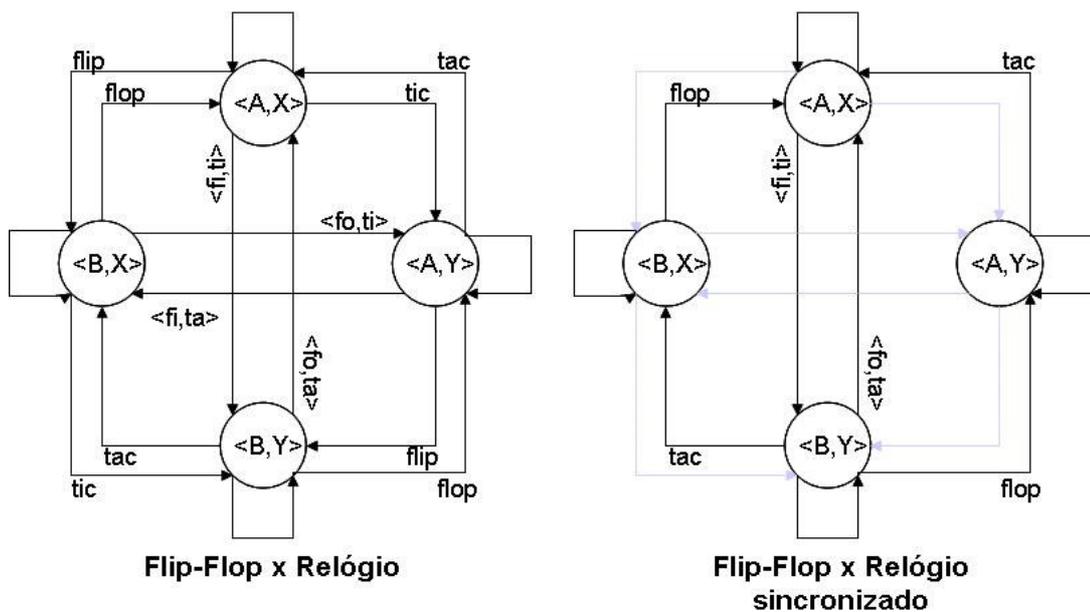


Figura 11 – Produto de sistemas e correspondente sincronização

Deste modo, a sincronização determina o comportamento conjunto de autômatos seqüenciais (MENEZES, 1996). É uma operação que consiste,

basicamente, de uma restrição da composição paralela, induzida por um conjunto de sincronizações (restrições) ao nível das etiquetas dos autômatos componentes.

3 Autômatos Temporizados

O modelo de Autômatos Temporizados (*Timed Automata*) de Alur e Dill foi proposto com o objetivo de modelar o comportamento de sistemas de tempo real. Tal modelo pode ser visto como um sistema de transição, o qual possui um grafo de transição finito com restrições de tempo associadas a suas arestas e estados (ALUR; DILL, 1994; ALUR, 1999).

As restrições são construídas a partir de variáveis de controle de tempo, chamadas de relógios. Um autômato temporizado pode representar tanto características quantitativas, como periodicidade, quanto características qualitativas, como o não determinismo. O modelo de tempo utilizado pelo autômato temporizado é denso, já que armazena valores reais para os relógios, os quais determinam o tempo decorrido desde a sua última inicialização. O incremento dos valores dos relógios é realizado de acordo com uma referência global de tempo. As transições entre os estados são consideradas instantâneas e o tempo passa durante a ocupação de um estado, também chamado de locação. Uma transição só pode ser realizada se a avaliação do relógio satisfizer a restrição da aresta. Do mesmo modo, um estado pode ser ocupado apenas pelo tempo tolerado pela restrição associada a ele, também chamada de restrição invariável.

A Fig. 12 exemplifica um autômato temporizado que possui os estados q_0 (estado inicial) e q_1 . Nesse exemplo, existe apenas um relógio rotulado como x , e duas restrições, uma associada à aresta (q_1, q_0) , e uma restrição invariável associada ao estado q_1 . A restrição ligada à aresta permite que ocorra uma transição de q_1 para q_0 apenas quando o valor do relógio x for superior ou igual a 1. Por outro lado, a restrição invariável em q_1 , restringe a permanência do sistema neste local apenas enquanto o relógio x possuir valores inferiores a 2.

Na aresta (q_0, q_1) existe uma inicialização de x ($x:=0$), a qual determina a atribuição de x com 0, toda vez que for utilizada. Desta maneira, o modelo expressa que um evento b só pode ocorrer dentro do intervalo de tempo $[1, 2]$ após a ocorrência de um a .

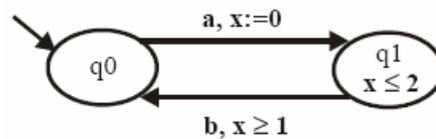


Figura 12 – Exemplo de um autômato temporizado

Formalmente, para um conjunto de variáveis de relógio X , o conjunto de restrições, denotado por $\Phi(X)$, possui as restrições de relógio φ definidas pela gramática: $\varphi = x \leq c \mid c \leq x \mid x < c \mid c < x \mid \varphi_1 \wedge \varphi_2$, onde x é um relógio em X e c é uma constante em \mathbb{Q} (qualquer valor racional não zero). Ou seja, as restrições são definidas a partir da comparação de um relógio com uma constante de tempo. Restrições como *verdadeiro*, $x = c$ e $x \in [2,5)$, podem ser definidas como formas abreviadas.

A interpretação de relógio v para um conjunto de relógios X , associa um valor real para cada relógio, ou seja, realiza o mapeamento de X para o conjunto \mathcal{R}^+ (números reais não negativos). Uma interpretação v para X satisfaz uma restrição de relógio φ sobre X , se e somente se, φ é verdadeira para os valores dados por v . Para $\delta \in \mathcal{R}$ (reais), $v + \delta$ denota a interpretação de relógio que mapeia todo relógio x para o valor $v(x) + \delta$. Para $Y \subseteq X$, $v[Y:=0]$ denota a interpretação de relógio para X que associa 0 para cada relógio $x \in Y$, e mantém a evolução dos outros relógios segundo v (ALUR; DILL, 1994; ALUR, 1999).

Definição 12 O autômato temporizado (ALUR, 1999) é definido como uma seis-upla:

$$Atemp = (L, L^0, \Sigma, X, I, E)$$

Onde:

L – conjunto finito de estados (ou locações);

L^0 – conjunto de estados iniciais ($L^0 \subseteq L$);

Σ – alfabeto de símbolos de entrada (ou rótulos);

X – conjunto finito de relógios;

I – mapeamento que associa cada estado s em L com alguma restrição de relógio em $\Phi(X)$, ou seja, uma restrição invariável;

E – conjunto de transições ($E \subseteq L \times \Sigma \times 2^X \times \Phi(X) \times L$). Uma transição $\langle s, a, \varphi, \lambda, s' \rangle$ representa uma transição do estado s para s' mediante o símbolo a . A restrição

de relógio φ determina quando a transição está habilitada, enquanto o conjunto $\lambda \subseteq X$ determina os relógios a serem inicializados nessa transição.

O comportamento do autômato temporizado pode ser definido a partir dos conceitos tradicionais de sistemas de transição. Seja $Atemp$ um autômato temporizado e S_{Atemp} um sistema de transição associado a ele, onde um estado de S_{Atemp} é um par (s, v) , tal que s é um estado de $Atemp$ e v é uma interpretação de relógio para X (conjunto finito de relógios) que satisfaz a restrição invariável $I(s)$. Um estado (s, v) é considerado inicial, se s é um estado de $Atemp$ e $v(x)=0$ para todos os relógios. O alfabeto de símbolos de entrada (Σ_{Atemp}) do sistema de transição S_{Atemp} é dado por $\Sigma \cup \mathcal{R}$ (reais). Há dois tipos de transições em S_{Atemp} , o estado pode mudar devido:

1) ao passar do tempo: para um estado (s, v) e um valor real de incremento de tempo $\delta \geq 0$, $(s, v) \xrightarrow{\delta} (s, v + \delta)$, se para todo $0 \leq \delta' \leq \delta$, $v + \delta'$ satisfaz a restrição invariável $I(s)$, ou seja, é tempo transcorrido no estado s até que a restrição invariável seja satisfeita;

2) à transição de estado no autômato temporizado: para um estado (s, v) e a uma transição $\langle s, a, \varphi, \lambda, s' \rangle$, tal que v satisfaz φ , $(s, v) \xrightarrow{a} (s', v[\lambda:=0])$, ou seja, é a transição do estado s para s' , com a leitura do símbolo do alfabeto de entrada a , em que seja satisfeita a restrição φ da aresta.

Por exemplo, o autômato da Fig. 12, definido como o autômato temporizado $Atemp = (\{q_0, q_1\}, \{q_0\}, \{a, b\}, \{x\}, I, \{(q_0, a, verdadeiro, \{x\}, q_1), (q_1, b, x \geq 1, \{\}, q_0)\})$, onde $I(q_1) = x \leq 2$, pode ser descrito pelo sistema de transição $S_{Atemp} = (Q_{Atemp}, Q_{Atemp}^0, \Sigma_{Atemp}, T)$, onde $Q_{Atemp} = \{q_0, q_1\} \times \mathcal{R}$ (reais), $Q_{Atemp}^0 = \{(q_0, 0)\}$, $\Sigma_{Atemp} = \{a, b\} \cup \mathcal{R}$ (reais) e T é conjunto de transições que respeitam os tipos definidos acima. A seqüência abaixo:

$$(q_0, 0) \xrightarrow{3} (q_0, 3) \xrightarrow{a} (q_1, 0) \xrightarrow{0,5} (q_1, 0.5) \xrightarrow{1} (q_1, 1.5) \xrightarrow{b} (q_0, 1.5) \xrightarrow{a} (q_1, 0)$$

é representada por uma série de transições válidas para o sistema S_{Atemp} e conseqüentemente descreve o comportamento do autômato temporizado $Atemp$ para uma determinada seqüência de eventos (ALUR, 1999).

Quando ocorrer a existência de mais de um relógio associado ao S_{Atemp} , o seu estado mapeia todos os relógios do sistema. Por exemplo, se for associado ao

autômato da Fig. 12 um outro relógio (y), então, seu estado seria definido por uma tripla $(s, v(x), v(y))$.

A seguir é apresentado um exemplo bastante relevante da aplicação de autômatos temporizados no modelo de Animação Gráfica baseada em Autômatos Temporizados Sincronizados (AGA-S), o qual é uma extensão do modelo AGA já citado na seção 2.5.

O modelo AGA-S estende o modelo AGA para prover restrições temporais às especificações dos atores e também ampliar as funcionalidades do modelo quanto à interação com o observador da animação. As restrições temporais são adicionadas ao modelo com o objetivo de atribuir propriedades de temporização ao padrão de comportamento dos atores, como também subsidiar a apresentação de mídias contínuas na animação, além das imagens estáticas utilizadas como saída no modelo AGA.

O elemento cênico (ator) é a unidade básica em uma animação AGA-S, seu objetivo é descrever as características estéticas e comportamentais de cada parte estrutural da animação. O ator AGA-S também tem o alfabeto de saída estendido para emitir não só objetos discretos¹, como no modelo AGA (imagens estáticas), mas também objetos contínuos, como, por exemplo, sons (ACCORSI; MENEZES; NEDEL, 2001).

Definição 13 *O ator AGA-S (ACCORSI; MENEZES; NEDEL, 2001) é definido como uma seis-upla:*

$$Ats = (Atemp, \Delta, \lambda_c, \sigma, F, D)$$

Onde:

$Atemp$ – autômato temporizado $Atemp = (L, L^0, \Sigma, X, I, E)$;

Δ – alfabeto de símbolos de saída;

λ_c – função de saída contextual (função parcial $\lambda_c: ExF^* \rightarrow \Delta^F$);

σ – função descrição (função parcial $\sigma: L \rightarrow D$);

F – conjunto finito de funções de transformação de Δ^F , em que Δ^F é composto por todas as imagens que podem ser produzidas pela composição de zero ou mais imagens de Δ transformadas por zero ou mais funções, ou seja, a imagem de saída

¹ O termo discreto indica que a apresentação do objeto é instantânea, enquanto o termo contínuo é adotado para objetos que necessitam de tempo para a apresentação.

pode sofrer transformações com o objetivo de se adequar ao contexto da cena representada;

D – conjunto finito de descrições.

As restrições temporais dos relógios em *Atemp* definem o comportamento dinâmico do ator em relação ao tempo de permanência nos estados e os intervalos disponíveis para transição. A Fig. 13 ilustra o ator *bicho*, apresentado inicialmente na seção 2.5, modelado a partir da definição do ator AGA-S. Nota-se que, agora, esse ator possui restrições temporais vinculadas aos estados e transições definidas sobre um relógio y .

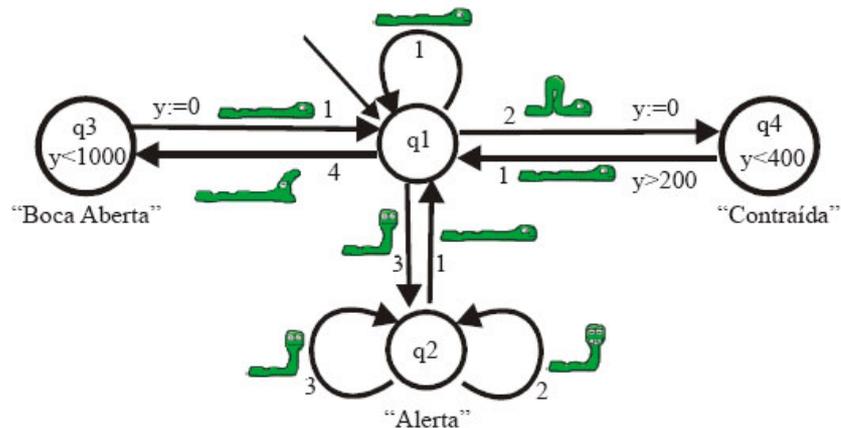


Figura 13 – Ator *bicho* temporizado

Ao estado $q4$ do ator *bicho* é atribuída uma restrição invariável, a qual determina que a imagem dele contraído será emitida, no máximo, por um tempo inferior a 400 milésimos de segundo, enquanto a restrição vinculada à aresta ($q4, q1$) determina que esta mesma imagem poderá ser apresentada, no mínimo, por um tempo inferior a 200 milésimos de segundo. Nota-se que, em toda fita de entrada especificada para esse ator, essas restrições deverão ser respeitadas e, portanto, todas as animações em que ele participar esse padrão de comportamento dinâmico será mantido.

O alfabeto de saída Δ é um conjunto finito de objetos para apresentação. Esses objetos podem ser discretos ou contínuos. Para esta definição do modelo AGA-S são estabelecidos quatro tipos primitivos de objetos: texto, som, gráfico (imagem) e vídeo.

Nota-se que a função de saída contextual λ_c permanece a mesma do modelo AGA, apenas foi adequada para vincular a saída às transições do autômato temporizado.

O conjunto de funções de transformação F , utilizado pela função de saída contextual, tem o mesmo propósito do definido no modelo original, porém, pode ser explorado para a definição de funções para tratamento de mídia contínua, como, por exemplo, selecionar parte da mídia para apresentação.

O comportamento de emissão da saída para os objetos discretos permanece o mesmo do modelo AGA. O objeto vinculado à aresta do autômato é transformado pelas funções determinadas na fita de entrada e apresentado no momento da transição. Para os objetos contínuos, em especial, a transição dispara o início da apresentação, e o tempo de duração da mídia avança durante a ocupação do estado.

A função descrição σ e o conjunto de descrições D mantêm as mesmas definições do modelo AGA com o propósito de estabelecer a vinculação de descrições semânticas aos estados do autômato temporizado.

É importante ressaltar, que embora as características da fita de entrada para os atores AGA-S permaneçam as mesmas do modelo AGA, durante a especificação do conteúdo das células, as restrições temporais devem ser respeitadas (ACCORSI; MENEZES; NEDEL, 2001).

3.1 Composição

A composição de autômatos temporizados é essencialmente uma composição de sistemas de transição, que além de todas as combinações de estados e transições, realiza também a junção das restrições e união dos relógios.

Formalmente, a composição de um autômato temporizado $Atemp_1 = (L_1, L_1^0, \Sigma_1, X_1, I_1, E_1)$ com $Atemp_2 = (L_2, L_2^0, \Sigma_2, X_2, I_2, E_2)$, denotado por $Atemp_1 || Atemp_2$, é o autômato temporizado $Atemp_{resultado} = (L_1 \times L_2, L_1^0 \times L_2^0, \Sigma_1 \cup \Sigma_2, X_1 \cup X_2, I, E)$, onde $I(s_1, s_2) = I(s_1) \wedge I(s_2)$. Assume-se que $X_1 \cap X_2 = \emptyset$. As transições de E são definidas por:

1) para $a \in \Sigma_1 \cap \Sigma_2$, para todo $\langle s_1, a, \varphi_1, \lambda_1, s'_1 \rangle$ em E_1 e $\langle s_2, a, \varphi_2, \lambda_2, s'_2 \rangle$ em E_2 , E possui $\langle (s_1, s_2), a, \varphi_1 \wedge \varphi_2, \lambda_1 \cup \lambda_2, (s'_1, s'_2) \rangle$.

2) para $a \in \Sigma_1 - \Sigma_2$, para todo $\langle s, a, \varphi, \lambda, s' \rangle$ em E_1 e todo t em L_2 , E possui $\langle (s, t), a, \varphi, \lambda, (s', t) \rangle$.

3) para $a \in \Sigma_2 - \Sigma_1$, para todo $\langle s, a, \varphi, \lambda, s' \rangle$ em E_2 e todo t em L_1 , E possui $\langle (t, s), a, \varphi, \lambda, (t, s') \rangle$.

Neste trabalho, a composição é realizada a partir de uma operação sintática onde não é efetuada a análise das restrições. Desta forma, as possíveis conjunções das restrições temporais que podem causar limites impossíveis de serem satisfeitos (restrições conflitantes) são solucionadas através de algoritmos de verificação de sistemas, conforme estão nos artigos estudados sobre autômatos temporizados (ALUR, 1994; ALUR; DILL, 1996; ALUR, 1999).

A composição de autômatos temporizados determina o sincronismo a partir dos símbolos comuns do alfabeto de entrada. Esta operação possibilita que sistemas complexos possam ser definidos a partir do produto dos sistemas componentes (ALUR, 1999).

Nesta seção é apresentado um exemplo sobre a composição entre uma cancela de bloqueio para passagem de trem e um controlador eletrônico que é sensível aos movimentos de chegada e partida do trem em frente à cancela.

O autômato temporizado $P = (\{t_0, t_1, t_2, t_3\}, \{t_0\}, \{abaixar, descida, levantar, subida\}, \{y\}, I_P, E_P)$ é criado para modelar o comportamento da cancela de bloqueio quanto ao tempo de resposta aos sinais de *abaixar* e *levantar*. O estado t_0 indica a cancela aberta (elevada), enquanto t_2 representa a cancela fechada (abaixada). Os estados t_1 e t_3 indicam a espera entre a requisição de uma tarefa e sua resposta com uma ação. As restrições invariáveis $I_P(t_1) = y \leq 1$ e $I_P(t_3) = y \leq 2$ em I_P determinam, respectivamente, que a cancela pode demorar no máximo 1 minuto depois de emitido o sinal *abaixar* para bloquear (*descida*), e pode esperar no máximo 2 minutos para abrir (*subida*) mediante o sinal de *levantar*. O conjunto E_P pode ser visto através da presença das arestas na Fig. 14. A restrição $y \geq 1$ na transição de t_3 para t_0 , indica o tempo de espera mínimo para determinar a subida da cancela. Devido a essa restrição e a restrição invariável em t_3 , após o sinal de *levantar*, o evento de *subida* deve ocorrer entre 1 e 2 minutos. Os controles temporais são realizados sobre o único relógio y .

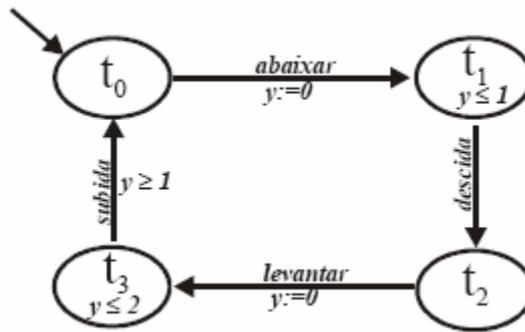


Figura 14 – Autômato temporizado da cancela

O autômato temporizado P se comunica com o controlador eletrônico através dos símbolos *abaixar* e *levantar*. A modelagem do comportamento do controlador é definida pelo autômato temporizado $C = (\{u_0, u_1, u_2\}, \{u_0\}, \{\text{aproximação, afastamento, abaixar, levantar}\}, \{z\}, I_C, E_C)$ quando este registra a aproximação e afastamento do trem. Os estados u_1 e u_2 simbolizam, respectivamente, o tempo de resposta para os eventos de aproximação e afastamento do trem. As restrições invariáveis $I_C(u_1) = I_C(u_2) = z \leq 1$ determinam que o tempo de resposta aos eventos deve ser de no máximo 1 minuto.

Devido à presença da restrição $z = 1$ na aresta de u_1 para u_0 , o tempo de resposta para abaixar a cancela fica fixado em exatamente 1 minuto. O conjunto de transições E_C pode ser visto através da presença das arestas na Fig. 15.

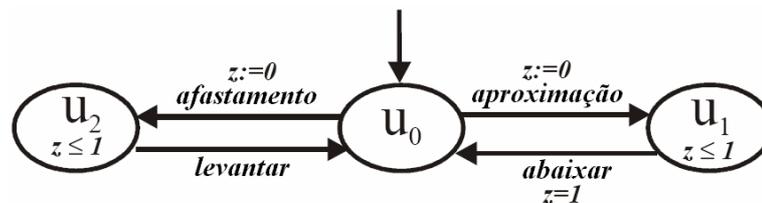


Figura 15 – Autômato temporizado do controlador eletrônico

A computação conjunta dos dois autômatos interagindo pode ser vista através da composição. Esta operação utiliza os símbolos comuns, *levantar* e *abaixar*, para estabelecer o sincronismo entre os dois autômatos. A composição temporizada dos autômatos é definida como:

$$PC = P||C = (L_{PC}, L_{PC}^0, \Sigma_{PC}, X_{PC}, I_{PC}, E_{PC})$$

Onde:

$$L_{PC} = \{t_0, t_1, t_2, t_3\} \times \{u_0, u_1, u_2\} = \{t_0u_0, t_0u_1, t_0u_2, t_1u_0, t_1u_1, t_1u_2, t_2u_0, t_2u_1, t_2u_2, t_3u_0, t_3u_1, t_3u_2\};$$

$$L_{PC}^0 = \{t_0\} \times \{u_0\} = \{t_0 u_0\};$$

$$E_{PC} = \{\text{abaixar, descida, levantar, subida}\} \cup \{\text{aproximação, saída, abaixar, levantar}\} = \{\text{aproximação, saída, abaixar, levantar, descida, subida}\};$$

$$X_{PC} = \{y\} \cup \{z\} = \{y, z\};$$

I_{PC} é o conjunto de restrições invariáveis obtido pela conjunção das restrições dos autômatos originais, por exemplo, $I_{PC}(t_1 u_1) = I_P(t_1) \wedge I_C(u_1) = y \leq 1 \wedge z \leq 1$. Os elementos desse conjunto podem ser observados em cada estado na Fig. 16.

E_{PC} é o conjunto de transições definidas de acordo com sua participação no autômato original. Para as transições de símbolos de sincronismo é preciso manter os estados de origem e destino originais, e compor a conjunção das restrições e inicializações. Por exemplo, a partir de $\langle t_0, \text{abaixar}, \{\}, \{y\}, t_1 \rangle$ em E_P e $\langle u_1, \text{abaixar}, \{z=1\}, \{\}, u_0 \rangle$ em E_C é produzida a transição $\langle t_0 u_1, \text{abaixar}, \{z=1\}, \{y\}, t_1 u_0 \rangle$. O conjunto completo pode ser observado através das arestas na Fig. 16.

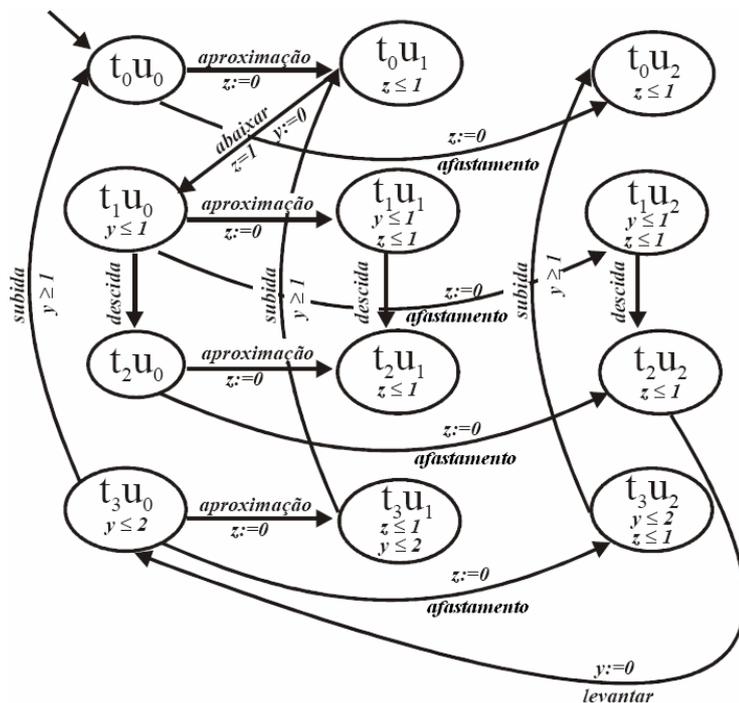


Figura 16 – Composição dos autômatos temporizados P (cancela) e C (controlador)

A Fig. 17 ilustra a seqüência de eventos $\alpha = (\text{aproximação}, 3)(\text{abaixar}, 4)(\text{descida}, 5)(\text{afastamento}, 9)(\text{levantar}, 10)(\text{subida}, 11.5)$, que respeita as restrições temporais da composição PC , é representada pelos pares (a, t) , onde $a \in \Sigma_{PC}$ e $t \in \mathcal{R}$ (reais). As caixas em cinza, na Fig. 17, demonstram a evolução do tempo para os relógios z e y , assim como suas inicializações. As linhas verticais indicam a ocorrência

4 Teoria das Categorias

Este capítulo tem a finalidade de apresentar os conceitos básicos e as principais operações de Teoria das Categorias a serem aplicadas no decorrer deste trabalho como uma ferramenta para o desenvolvimento do sincronismo entre os autômatos.

A Teoria das Categorias, criada por Samuel Eilenberg e Saunders Mac Lane em 1945, é uma teoria recente que traz consigo uma idéia inovadora de entendimento e abordagem. É um fato bem conhecido que essa teoria é considerada atualmente uma das mais valiosas ferramentas em Computação Científica, especialmente na área de Informática Teórica. Entre as diversas características que motivam seu uso em Ciência da Computação, destacam-se:

- *independência da implementação*: permite formalizar propriedades abstratas independentes de estruturas;

- *dualidade*: a noção categorial de dualidade divide o trabalho pela metade, tanto em termos de construções como de resultados;

- *herança de resultados e comparação de expressividade de formalismos*: onde a primeira permite que se construam novas categorias a partir de categorias existentes, simplificando sobremaneira e até dispensando a verificação de alguns resultados, e a segunda é possível comparar a passagem de um tipo de estrutura matemática para outra;

- *notação gráfica*: é a expressão de equações na forma de diagramas, facilitando a identificação e a compreensão de seus componentes e de seus relacionamentos;

- *expressividade de suas construções*: permite formalizar idéias complexas de forma simples, bem como propicia um novo, ou melhor entendimento das questões relacionadas a toda Ciência da Computação (HAEUSLER; MENEZES, 2001; DIRETRIZES, 2005). Essas construções freqüentemente não possuem paralelo nas demais teorias como na Teoria de Conjuntos (HAEUSLER; MENEZES, 2001).

Teoria das Categorias estuda “objetos” (conjuntos estruturados) e

“morfismos” (setas) entre eles. Estas entidades são ambas primitivas em Teoria das Categorias. Essa é uma abordagem diferente da usualmente adotada na Teoria dos Conjuntos, onde os elementos e as funções são as entidades primitivas, e não os objetos e morfismos.

Assim, propriedades, na Teoria das Categorias, devem ser especificadas em termos dos objetos e morfismos, ou seja, em termos de existência, unicidade, equações, etc., definidas sobre objetos e setas (ASPERTI; LONGO, 1991).

A seguir é apresentado o conceito de Categoria no qual, além das coleções de objetos e morfismos, é introduzida uma operação denominada composição (de morfismos), a qual deve ser associativa. E adicionalmente, a cada objeto deve ser associado um morfismo identidade (MAC LANE, 1971).

4.1 Categoria

Definição 14 *Uma categoria C é uma seis-upla:*

$$C = \langle Ob_C, Mor_C, \partial_0, \partial_1, \iota, o \rangle$$

Onde:

Ob_C – uma coleção de objetos, denotados por a, b, \dots, A, B, \dots ;

Mor_C – uma coleção de morfismos, denotados por f, g, \dots ;

$\partial_0, \partial_1: Mor_C \rightarrow Ob_C$ – são operações denominadas Domínio (origem) ou Codomínio (destino), respectivamente. Um morfismo f tal que $\partial_0(f) = A$ (ou $dom(f) = A$) e $\partial_1(f) = B$ (ou $cod(f) = B$) é usualmente denotado por:

$$f: A \rightarrow B$$

$\iota: Ob_C \rightarrow Mor_C$ – uma operação denominada identidade (id) tal que cada objeto b é associado a um morfismo:

$$\iota_b: b \rightarrow b$$

$o: (Mor_C)^2 \rightarrow Mor_C$ – uma operação denominada composição tal que cada par de morfismos:

$$\langle f: a \rightarrow b, g: b \rightarrow c \rangle$$

É associado a um morfismo:

$$g \circ f: a \rightarrow c$$

tal que $dom(f \circ g) = dom(g)$ e $cod(f \circ g) = cod(f)$.

As operações de *identidade* e *composição* devem satisfazer as seguintes propriedades:

a) Propriedade da Identidade: para quaisquer setas f e g , tal que $\text{cod}(f) = \text{dom}(g) = b$, tem-se que: $id_b \circ f = f$ e $g \circ id_b = g$.

b) Propriedade Associativa: para quaisquer setas f , g e h , tal que $\text{dom}(f) = \text{cod}(g)$ e $\text{dom}(g) = \text{cod}(h)$, tem-se que: $(f \circ g) \circ h = f \circ (g \circ h)$.

Observações:

– Dados dois objetos A e B , o conjunto de todos os morfismos f tal que $f: A \rightarrow B$ é denotado por $C[A, B]$. Assim, $f \in C[A, B]$ significa que $\text{dom}(f) = A$ e $\text{cod}(f) = B$.

– Pela definição de categoria, todo objeto $B \in Ob_C$ deve possuir uma identidade $id_B: B \rightarrow B$, e esta identidade é única pois se existisse uma $id_{B'} \neq id_B$ então, pela propriedade da identidade, $id_{B'} = id_B \circ id_{B'} = id_B$.

– Um morfismo em que coincide origem e destino é chamado de endomorfismo.

Em Teoria das Categorias não consideramos a estrutura interna dos objetos, apenas as relações estabelecidas entre eles pelos seus morfismos (MENEZES, 1996).

Na Tab. 1 segue uma lista de alguns exemplos de categorias especificando seus objetos e morfismos (ASPERTI; LONGO, 1991; MENEZES, 1996):

Tabela 1 – Exemplos de categorias

Categoria	Objetos	Morfismos
Set	Conjuntos	Funções (totais)
Top	Espaços topológicos	Funções contínuas
Vect	Espaços vetoriais	Transformações lineares
Grp	Grupos	Homomorfismos de grupos
PO	Conjuntos parcialmente ordenados	Funções monotônicas

Por exemplo, para comprovar que **Set** é uma categoria, basta verificar a associatividade de funções e a identidade:

Dados os morfismos $f: A \rightarrow B$, $g: B \rightarrow C$ e $h: C \rightarrow D$, e para qualquer $a \in A$:

$$((h \circ g) \circ f)(a) = (h \circ g)(f(a)) = h(g(f(a))) = h(g \circ f(a)) = (h \circ (g \circ f))(a)$$

Para qualquer $f: A \rightarrow B$ e para qualquer $a \in A$:

$$(f \circ id_A)(a) = f(id_A(a)) = f(a) = b = id_B(b) = id_B(f(a)) = (id_B \circ f)(a)$$

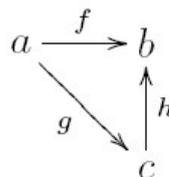
4.2 Diagramas

Uma importante ferramenta da aplicação de Teoria das Categorias é o uso de *diagramas* para representar equações (ASPERTI; LONGO, 1991), facilitando sobremaneira a identificação e a compreensão de suas componentes e de seus relacionamentos. E ainda permite inferir novas equações a partir de outras já conhecidas. Além disso, é essencial ressaltar o uso dos diagramas de setas, pois muitas propriedades de sistemas matemáticos podem ser unificadas e simplificadas por eles (MAC LANE, 1971).

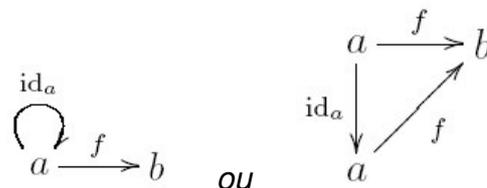
Definição 15 Em um diagrama, o morfismo $f \in C[a, b]$ é desenhado como uma seta do objeto a para o objeto b rotulada por f .

$$a \xrightarrow{f} b$$

Exemplo 4 Sejam os morfismos $f: a \rightarrow b$, $g: a \rightarrow c$ e $h: c \rightarrow b$, então seu diagrama é:

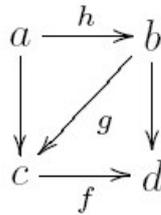


E para representação da identidade seria:

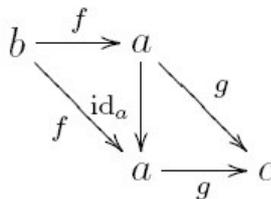


Um diagrama é dito um *diagrama comutativo* se, a composição dos morfismos ao longo de qualquer caminho entre dois objetos fixos é igual. Por exemplo, as propriedades de associatividade e identidade da definição de categoria podem ser adequadamente visualizadas pelos diagramas comutativos que seguem (ASPERTI; LONGO, 1991):

- Propriedade Associativa: Dados os morfismos $f: c \rightarrow d$, $g: b \rightarrow c$ e $h: a \rightarrow b$, então $(f \circ g) \circ h = f \circ (g \circ h)$.



- Propriedade Identidade: Dados os morfismos $f: b \rightarrow a$, $g: a \rightarrow c$, então $id_a \circ f = f$ e $g \circ id_a = g$.



4.3 Categoria dos Grafos

Grafos são freqüentemente usados para representar sistemas. Alguns exemplos típicos foram mencionados como os formalismos: autômatos e sistemas de transição. Assim tal importância nesses casos, são as construções categoriais de produto e coproduto que podem ser interpretadas como combinadores de sistemas (HAEUSLER; MENEZES, 2001).

Resumidamente, um grafo é constituído de nodos e arcos. A cada arco é associado um nodo origem e um nodo destino (DIESTEL, 2005; FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2005).

Faz-se necessário especificar o conceito categorial de grafos, sendo que serão utilizadas a operação categorial de produto e demais construções categoriais estudadas nessa monografia.

Definição 16 Um grafo G é uma quatro-upla:

$$G = \langle V, T, \partial_0, \partial_1 \rangle$$

Onde:

V – um conjunto de nodos ou vértices;

T – um conjunto de arcos ou arestas;

$\partial_0, \partial_1: T \rightarrow V$ – são operações totais (funções) denominadas origem e destino, respectivamente. Um arco normalmente é denotado por $t: A \rightarrow B$, tal que $\partial_0(t) = A$ e $\partial_1(t) = B$.

4.3.1 Homomorfismo de grafos

Um homomorfismo de grafos deve preservar a estrutura dos grafos, ou seja, ao mapear os nodos e arcos, e deve preservar as operações de origem e destino. Uma interpretação usual para um homomorfismo de grafos é a de *implementação* (simulação), ou seja, como o grafo origem pode ser implementado (simulado) pelo destino. Assim, se grafos representam sistemas, um homomorfismo de grafos representa uma implementação de um sistema em outro.

Definição 17 Sejam $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ grafos. Um homomorfismo de grafos:

$$h: G_1 \rightarrow G_2$$

é um par de funções:

$$h = \langle h_V, h_T \rangle$$

onde:

$$h_V: V_1 \rightarrow V_2 \quad e \quad h_T: T_1 \rightarrow T_2$$

são tais que:

$$\partial_{02} \circ h_T = h_V \circ \partial_{01} \quad e \quad \partial_{12} \circ h_T = h_V \circ \partial_{11}$$

Exemplo 5 Dados os grafos:

$$G_1 = \langle \{A, B, C\}, \{r, s, t\}, \partial_{01}, \partial_{11} \rangle$$

$$G_2 = \langle \{1, 2, 3\}, \{a, b, c, d, e\}, \partial_{02}, \partial_{12} \rangle$$

Ilustrados na Fig. 18. O homomorfismo de grafos: $h = \langle h_V, h_T \rangle: G_1 \rightarrow G_2$ ilustrado na mesma figura preserva origem e destino dos arcos (HAEUSLER; MENEZES, 2001).

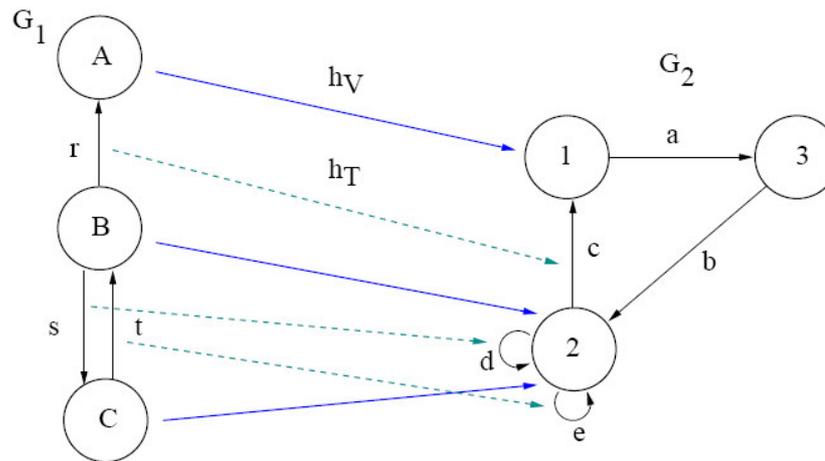


Figura 18 – Homomorfismo de grafos preserva origem e destino

4.3.2 Categoria Gr

A categoria Gr dos grafos é tal que possui todos os grafos como objetos e todos os homomorfismos de grafos como setas. Sendo que a composição de homomorfismos de grafos é definida a partir da composição das funções componentes.

Definição 18 A categoria Gr , definida como segue, constitui uma categoria:

$$Gr = \langle Ob_{Gr}, Mor_{Gr}, \partial_0, \partial_1, \iota, o \rangle$$

Onde:

Ob_{Gr} – coleção de todos os grafos;

Mor_{Gr} – coleção de todos os homomorfismos de grafos;

$\partial_0, \partial_1: Mor_{Gr} \rightarrow Ob_{Gr}$ – são tais que, para qualquer homomorfismo de grafos h :

$G_1 \rightarrow G_2$ onde $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$, tem-se que:

$$\partial_0(h) = G_1 \quad e \quad \partial_1(h) = G_2$$

$\iota: Ob_{Gr} \rightarrow Mor_{Gr}$ – é tal que cada grafo G em que $G = \langle V, T, \partial_0, \partial_1 \rangle$ é associado ao homomorfismo identidade:

$$id_G: \langle id_V, id_T \rangle = G \rightarrow G$$

em que: $id_V: V \rightarrow V$ e $id_T: T \rightarrow T$ são as funções identidade, e claramente id_G é um homomorfismo de grafos e satisfaz a propriedade da identidade.

$o: (Mor_{Gr})^2 \rightarrow Mor_{Gr}$ – uma operação parcial de composição definida sobre as funções componentes dos homomorfismos de grafos, ou seja, para quaisquer $f = \langle f_V, f_T \rangle: G_1 \rightarrow G_2$ e $g = \langle g_V, g_T \rangle: G_2 \rightarrow G_3$, tem-se que:

$$g \circ f = \langle g_V, g_T \rangle \circ \langle f_V, f_T \rangle = \langle g_V \circ f_V, g_T \circ f_T \rangle$$

Como a composição de funções é associativa, então a composição de homomorfismos de grafos também é associativa.

4.4 Produto Categorial

Pode-se dizer que o Produto em Teoria das Categorias generaliza a noção de Produto Cartesiano da Teoria dos Conjuntos. Já o Coproduto ou Soma generaliza a União Disjunta da Teoria dos Conjuntos. Vale a pena ressaltar que em Teoria das Categorias existe o conceito de dualidade, que “divide o trabalho pela metade, como também correlaciona conceitos (aparentemente distintos)”, e Produto e Coproduto são conceitos duais (ASPERTI; LONGO, 1991). Nesta monografia abordaremos somente o produto categorial para realização da composição paralela de autômatos, de acordo com a definição abaixo:

Definição 19 *Sejam C uma categoria e $a, b \in Ob_C$. O produto categorial de a e b é composto por um objeto $a \times b$ junto com dois morfismos (projeções) $p_a: a \times b \rightarrow a$ e $p_b: a \times b \rightarrow b$ tal que, dado um objeto $c \in Ob_C$ e para quaisquer morfismos $f \in C[c, a]$ e $g \in C[c, b]$, existe exatamente um único morfismo $h \in C[c, a \times b]$ tal que o diagrama ilustrado na Fig. 19 comuta.*

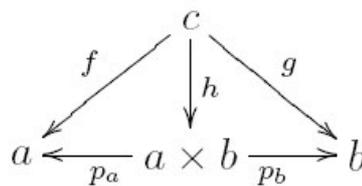


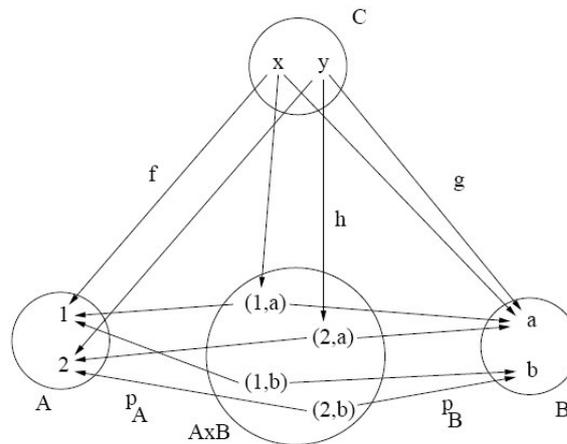
Figura 19 – Diagrama comutativo para produtos

Portanto, um produto categorial não é um objeto, mas sim uma tripla constituída por um objeto e dois morfismos: $\langle a \times b, p_a, p_b \rangle$.

Qualquer objeto C juntamente com quaisquer morfismos $f: c \rightarrow a$ e $g: c \rightarrow b$ é usualmente denominado de *pré-produto* dos objetos a e b . Logo, o produto $\langle a \times b, p_a, p_b \rangle$ também é um pré-produto.

Em vista disso, assim como o produto cartesiano, o produto categorial é uma operação reversível. Ou seja, em um produto $\langle a \times b, p_a, p_b \rangle$ os dois morfismos de projeção p_a e p_b permitem recuperar as componentes originais do produto (MAC LANE, 1971; ASPERTI; LONGO, 1991).

Exemplo 6 Em **Set**, sejam $A = \{1, 2\}$, $B = \{a, b\}$ e $C = \{x, y\}$. Então $A \times B = \{(1, a), (2, a), (1, b), (2, b)\}$ e h é o único morfismo que faz o diagrama comutar. Definimos f, g e h de acordo com a seguinte ilustração abaixo:



4.5 Morfismos

Considerando-se que, em Teoria das Categorias, objetos são entidades primitivas, não é possível especificar construções fundamentais – que são expressas em termos de objetos e morfismos – analisando suas estruturas internas. Dessa forma, as construções devem ser especificadas em termos de morfismos.

Os seguintes tipos de morfismos são introduzidos (CAMPANI, 2005):

- Monomorfismos, que generaliza a noção de função injetora.
- Epimorfismo, que generaliza a noção de função sobrejetora.
- Isomorfismo, que generaliza a noção de função bijetora.

Monomorfismo: Um morfismo $h: a \rightarrow b$ é dito *monomorfismo* ou simplesmente *mono*, tal que para qualquer objeto c e para quaisquer morfismos $g: c \rightarrow a$ e $f: c \rightarrow a$, se o diagrama na Fig. 20 comuta, então tem-se que: $g = f$.

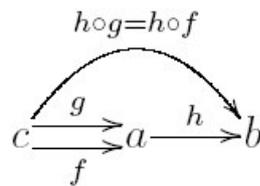


Figura 20 – Diagrama comutativo para monomorfismo

Epimorfismo: Um morfismo $h: a \rightarrow b$ é dito *epimorfismo* ou simplesmente *epi*, tal que para qualquer objeto c e para quaisquer morfismos $g: b \rightarrow c$ e $f: b \rightarrow c$, se o diagrama na Fig. 21 comuta, então tem-se que: $g = f$.

É o conceito dual de monomorfismo, obtido pela inversão de setas na definição.

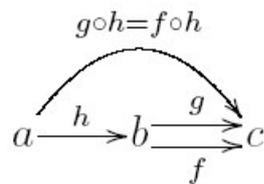


Figura 21 – Diagrama comutativo para epimorfismo

Isomorfismo: Um morfismo $h: a \rightarrow b$ é dito *isomorfismo* ou simplesmente *iso* conforme a Fig. 22, se e somente se h possui um *morfismo inverso*, ou seja, se existe $g: b \rightarrow a$, tal que: $g \circ h = id_a$ e $h \circ g = id_b$. É importante ressaltar que todo isomorfismo é *mono* e *epi*, embora o contrário não seja necessariamente verdadeiro.

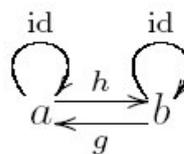


Figura 22 – Diagrama de um isomorfismo

4.6 Sub-Objeto

Sub-objeto é a versão categorial de subconjunto da Teoria dos Conjuntos. Baseia-se na idéia de definir um subconjunto $A \subseteq B$ como um monomorfismo $f: D \rightarrow B$ (intuitivamente, um monomorfismo f tal que " $f(D) = A$ ").

Se existem muitas setas *mono* definindo o mesmo subconjunto, é necessário introduzir uma classe de equivalência e definir sub-objetos usando-a.

Definição 20 *Seja C uma categoria. Se $f: b \rightarrow a$ e $g: c \rightarrow a$ são duas setas mono com destinos comuns em a , então dizemos que $f \leq g$ (g fatora f ou, f é fatorado por g), se e somente se existe $h: b \rightarrow c$ tal que $g \circ h = f$.*

Nota-se que o único h é mono também, pois:

$$h \circ k = h \circ k' \Rightarrow g \circ h \circ k = g \circ h \circ k' \Rightarrow f \circ k = f \circ k' \Rightarrow k = k'.$$

Se $f \leq g$ e $g \leq f$, então dizemos que $f \cong g$. Então \cong é uma relação de equivalência entre monomorfismos com destino comum. As classes de equivalência determinadas por esta relação de equivalência são os sub-objetos de a (ASPERTI; LONGO, 1991; MENEZES, 1996).

4.7 Equalizador

É uma construção categorial a qual, embora possa ser interpretada adequadamente em **Set**, não possui correspondentes operações padrões na Teoria dos Conjuntos capazes de expressá-las (MAC LANE, 1971; CAMPANI, 2005).

Dado um diagrama D constituído por dois morfismos paralelos $f, g: A \rightarrow B$, um equalizador pode ser interpretado, resumidamente, como segue:

O sub-objeto de A que caracteriza exatamente as condições em que f e g são iguais. Assim, como o próprio nome indica, um equalizador é uma operação que calcula a estrutura maximal, sem redundâncias, para a qual as duas setas paralelas sempre coincidem.

Exemplo 7 *Conforme a Fig. 23 abaixo, sejam $f, g: A \rightarrow B$ um par de funções “paralelas” em **Set**. O subconjunto $E \subseteq A$ no qual f e g coincidem é chamado equalizador de f e g . Em uma caracterização categorial, E deve ser representado por um sub-objeto, como uma seta mono $i: E \rightarrow A$ e i deve satisfazer a composição: $f \circ i = g \circ i$.*

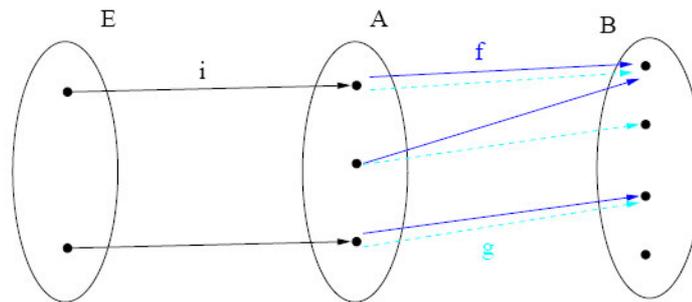


Figura 23 – Exemplo de equalizador em Set

Para garantir que i represente o sub-objeto “maximal” devemos adicionalmente exigir que:

- se $h: C \rightarrow A$ é outra função, tal que $f \circ h = g \circ h$, então h “fatora” de forma única por meio de i , ou seja, existe apenas um $k: C \rightarrow E$, tal que $h = i \circ k$. Como é representado na Fig. 24.

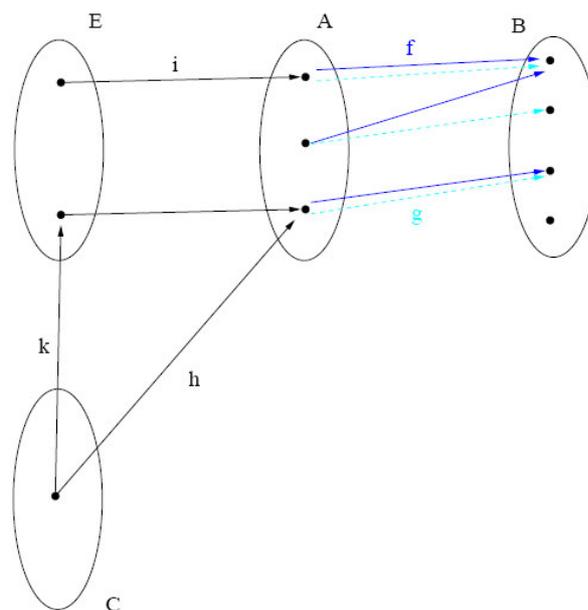


Figura 24 – Exemplo de equalizador em Set com i sendo o sub-objeto maximal

Desde que essa condição de fatoração única também implica que i é *mono*, é fornecida a definição a seguir:

Definição 21 Dado um par de morfismos $f, g \in C[a, b]$, um equalizador de f e g é um par (e, i) , $e \in Ob_C$ e $i \in C[e, a]$, conforme a Fig. 25, tal que:

- $f \circ i = g \circ i$;

- Para todo $h \in C[c, a]$, $f \circ h = g \circ h$ implica que existe um único morfismo $k \in C[c, e]$, tal que $i \circ k = h$.

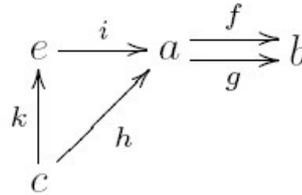


Figura 25 – Diagrama comutativo para equalizador

O morfismo h é chamado de *pré-equalizador*. E o conceito dual de equalizador é o co-equalizador (basta inverter os morfismos).

Como introduzido anteriormente na subseção 4.3.1, uma interpretação usual para um homomorfismo de grafos é a de *implementação* (simulação), ou seja, como o grafo origem pode ser implementado (simulado) pelo destino. Assim, se grafos representam sistemas, um homomorfismo representa uma implementação de um sistema em outro. Nesse contexto, o grafo resultante do cálculo do equalizador de duas implementações paralelas (dois homomorfismos paralelos) é a parte do sistema origem que é igualmente implementada no sistema destino. Ou seja, identifica o subsistema que é implementado da mesma forma pelos dois homomorfismos (MAC LANE, 1971; CAMPANI, 2005).

A definição abaixo, é uma das noções mais poderosas de Teoria das Categorias: o produto fibrado (*pullback*). Onde queremos generalizar os equalizadores para morfismos com diferentes origens.

4.8 Produto Fibrado

O *produto fibrado* constitui uma construção categorial importante e poderosa, resumidamente pode ser interpretado como segue:

Dados dois morfismos $f: b \rightarrow a$ e $g: c \rightarrow a$ (com um mesmo objeto destino), um produto fibrado é um *produto categorial* dos objetos b e c com dependência em a , ou um *produto* de b e c equalizado por f e g (MENEZES, 1996).

Definição 22 *Sejam C uma categoria e um diagrama D em C constituído por um par de morfismos $f: b \rightarrow a$ e $g: c \rightarrow a$ com destino comum em a , o produto fibrado*

de (f, g) é um objeto $b \times_a c$ e duas setas $p: b \times_a c \rightarrow b$ e $q: b \times_a c \rightarrow c$, conforme a Fig. 26, tal que:

- $f \circ p = g \circ q$, onde $f \circ p, g \circ q: b \times_a c \rightarrow a$;
- Para qualquer outra tripla $(d, h: d \rightarrow b, k: d \rightarrow c)$, tal que $g \circ k = f \circ h$, existe uma única seta $\langle h, k \rangle_a: d \rightarrow b \times_a c$, tal que $p \circ \langle h, k \rangle_a = h$ e $q \circ \langle h, k \rangle_a = k$, que faz o diagrama comutar.

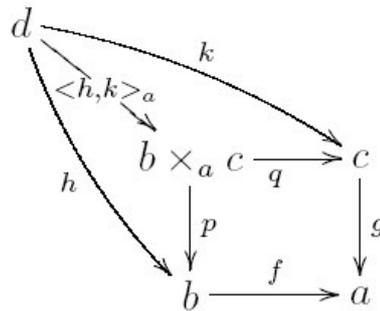


Figura 26 – Diagrama comutativo para produto fibrado

Observações:

- Os subscritos “a” indicam dependência de $b \times_a c$ e $\langle h, k \rangle_a$ sobre f e g , mas os subscritos não são opcionais, pois a notação deve indicar quando se trata de um produto fibrado.
- O “quadrado” de baixo é chamado de “quadrado do produto fibrado” (*pullback square*).
- Uma característica notável é a semelhança entre o produto fibrado e o produto, onde na verdade o produto é um caso particular de produto fibrado (HAEUSLER; MENEZES, 2001; CAMPANI, 2005).

4.8.1 Exemplo do Produto Fibrado como Restrição em Set

Na categoria **Set**. A construção do produto fibrado pode ser usada para calcular a restrição de uma função a partir de um subconjunto de seu codomínio. De fato, sejam $f: A \rightarrow B$ uma função e $B' \subseteq B$. Como B' é subconjunto de B , existe uma função inclusão $inc: B' \rightarrow B$ é *mono*. Então a função $f': A' \rightarrow B'$, calculada pelo produto fibrado ilustrado na Fig. 27, é a restrição de f para o codomínio B' , ou seja na Fig. 28:

para qualquer $a \in A'$, tem-se que $f'(a) = f(a)$.

A Fig. 29 ilustra um caso particular de produto fibrado como restrição em **Set**.

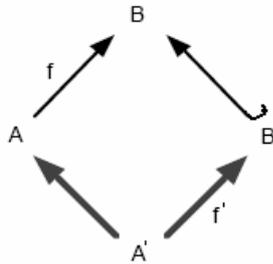


Figura 27 – Produto fibrado como cálculo de uma restrição

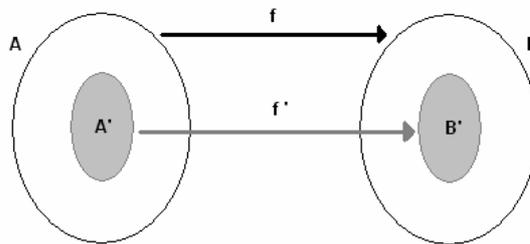


Figura 28 – Restrição de uma função

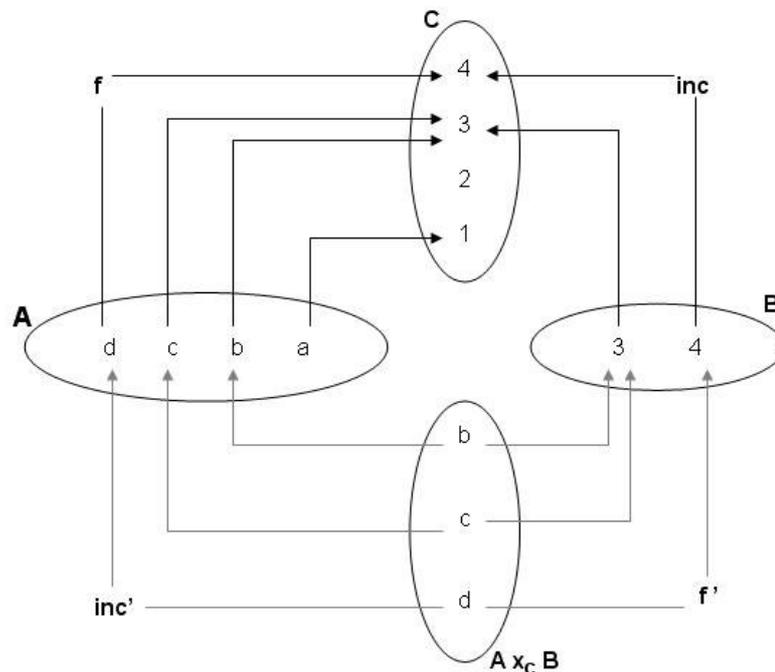


Figura 29 – Exemplo de restrição de uma função

Analogamente ao produto, o produto fibrado pode ser usado como operador de sistemas, onde a idéia básica consiste em restringir um sistema, de forma a satisfazer determinadas condições. A restrição em questão é obtida como uma generalização da restrição em **Set**.

A generalização da restrição em **Set** para grafos pode ser interpretada como um operador (unário) que permite obter uma *restrição de sistema*, ou seja, uma parte do sistema original, preservando as suas características (MENEZES, 1996; HAEUSLER; MENEZES, 2001). Essa abordagem pode ser usada para, por exemplo:

- Reuso ou Herança: o reuso ou herança de “código”, ou seja, o aproveitamento (reuso) de parte de um sistema existente, como alguns tipos de herança na programação orientada a objetos.
- Sincronização: a sincronização de sistemas de acordo com uma tabela de sincronizações (restrições).

Sendo que esse último item é o principal para o desenvolvimento da contribuição desta monografia, na qual pretende-se sincronizar autômatos temporizados e não temporizados com a construção categorial de produto fibrado como restrição.

5 Abordagem Categórica de Sincronização de Autômatos

5.1 Sincronização de Autômatos Não Temporizados

A idéia principal deste trabalho é desenvolver uma abordagem categórica de sincronização de autômatos com o objetivo de prover restrições em sistemas, e de poder sincronizar determinadas ações de um sistema componente com outro, e vice-versa. As aplicações da utilização de autômatos, já citadas no decorrer da monografia, são inúmeras. Nesta seção escolhemos um importante exemplo baseado nos atores do modelo AGA que compõem uma animação, já descritos na seção 2.5, para exemplificar as operações categóricas utilizadas para o sincronismo dos autômatos.

De acordo com as construções categóricas estudadas anteriormente, é possível formalizarmos categorialmente o produto categórico e o produto fibrado baseado em grafos.

Definição 23 *Sejam $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ grafos. O produto dos grafos G_1 e G_2 é:*

*Um grafo $G_1 \times G_2 = \langle V_1 \times V_2, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle$ – onde $\partial_{k1} \times \partial_{k2}$, para $k \in \{0, 1\}$, são induzidos unicamente pelo produto em **Set** dos vértices e arestas como ilustrado na Fig. 30 (é notável que $\langle T_1 \times T_2, \partial_{k1} \circ \pi_{T1}, \partial_{k1} \circ \pi_{T2} \rangle$ é pré-produto de $\langle V_1 \times V_2, \pi_{V1}, \pi_{V2} \rangle$ – juntamente com os morfismos $\pi_1 = \langle \pi_{V1}, \pi_{T1} \rangle: G_1 \times G_2 \rightarrow G_1$ e $\pi_2 = \langle \pi_{V2}, \pi_{T2} \rangle: G_1 \times G_2 \rightarrow G_2$, conforme são mostrados na Fig. 30.*

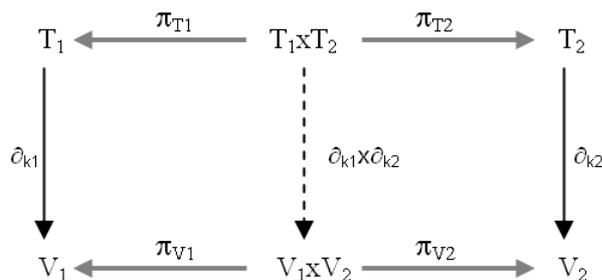


Figura 30 – Diagrama comutativo para produto de grafos

Definição 24 Considerando a categoria Gr dos grafos. Seja D um diagrama constituído pelos homomorfismos de grafos $f = \langle f_V, f_T \rangle: G_2 \rightarrow G_1$ e $g = \langle g_V, g_T \rangle: G_3 \rightarrow G_1$ (um mesmo objeto destino), onde $G_j = \langle V_j, T_j, \partial_{0j}, \partial_{1j} \rangle$, para $j \in \{1, 2, 3\}$. Então um produto fibrado de D , conforme a Fig. 31, é dado por:

$$\langle G_2 \times_{G_1} G_3, \langle p_V, p_T \rangle, \langle q_V, q_T \rangle \rangle$$

Onde o objeto resultante é constituído como segue:

$$G_2 \times_{G_1} G_3 = \langle V_2 \times_{V_1} V_3, T_2 \times_{T_1} T_3, \partial_0, \partial_1 \rangle$$

Em que ∂_k , para $k \in \{0, 1\}$, são induzidos unicamente pelo produto fibrado em **Set** dos vértices e arestas, ou seja, $\partial_k: T \rightarrow V$ são operações totais (funções) denominadas origem ($\partial_0 = T_2 \times_{T_1} T_3$) e destino ($\partial_1 = V_2 \times_{V_1} V_3$), respectivamente; considerando que $\langle T_2 \times_{T_1} T_3, p_T \circ \partial_{k2}, q_T \circ \partial_{k3} \rangle$ é pré-produto fibrado de $\langle V_2 \times_{V_1} V_3, p_V, q_V \rangle$.

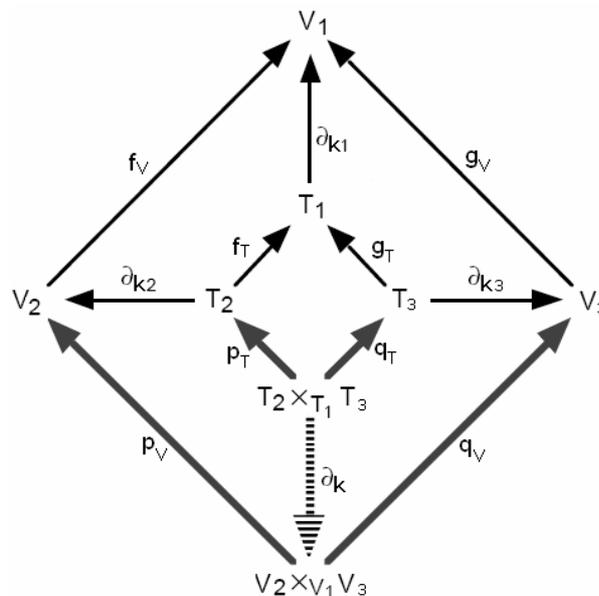


Figura 31 – Diagrama comutativo para produto fibrado de grafos

Todos os exemplos baseados em grafos, conforme a definição da categoria Gr , está sendo modificada para grafos rotulados e com seus nodos distinguidos para melhor representação dos sistemas envolvidos neste trabalho.

Com base no exemplo especificado na subseção 4.8.1, a seguir será proporcionado todo o processo de sincronização de autômatos (sistemas) com as restrições que se deseja, conforme a possibilidade da generalização da restrição em **Set** para grafos.

Tomando-se como base dois grafos (sistemas), pode-se aplicar a operação categorial de produto de grafos. O estado inicial dos autômatos a serem utilizados serão deixados de forma indefinida, isto significa dizer que qualquer um dos estados poderá ser considerado estado inicial. A escolha de qual dos estados será o inicial poderá ser feita pela própria máquina, que escolhe aleatoriamente qual dos estados será considerado inicial, ou pelo usuário que define a partir de que ponto deseja iniciar a seqüência. A Fig.32 representa os dois autômatos, G_1 simbolizando o autômato *Bicho* e G_2 simbolizando o autômato *Maçã*, que compõem uma suposta animação. Onde uma animação é vista, conforme a modelagem de um ator no modelo AGA, como uma lista de autômatos sincronizados, modelados – nos exemplos especificados – através de máquinas de Mealy, em que cada autômato contém sua própria fita de entrada, que irá determinar seu comportamento. Em que o alfabeto de saída é formado por objetos gráficos e mídias sonoras, do qual serão formadas palavras de saída que irão descrever a animação do ator.

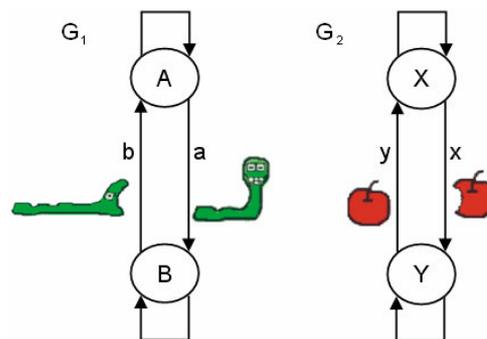


Figura 32 – Autômatos de uma animação

A Fig. 33 ilustra o produto dos grafos aplicado aos dois autômatos da Fig. 32. No objeto resultante do produto, as transições são pares de transições dos grafos componentes, como na categoria de grafos. Porém, quando visto como o comportamento conjunto de sistemas, pode-se interpretar de forma completamente diferente, pois reflete todas as combinações possíveis, síncronas e assíncronas das transições dos grafos originais, conforme ilustra a Fig. 33.

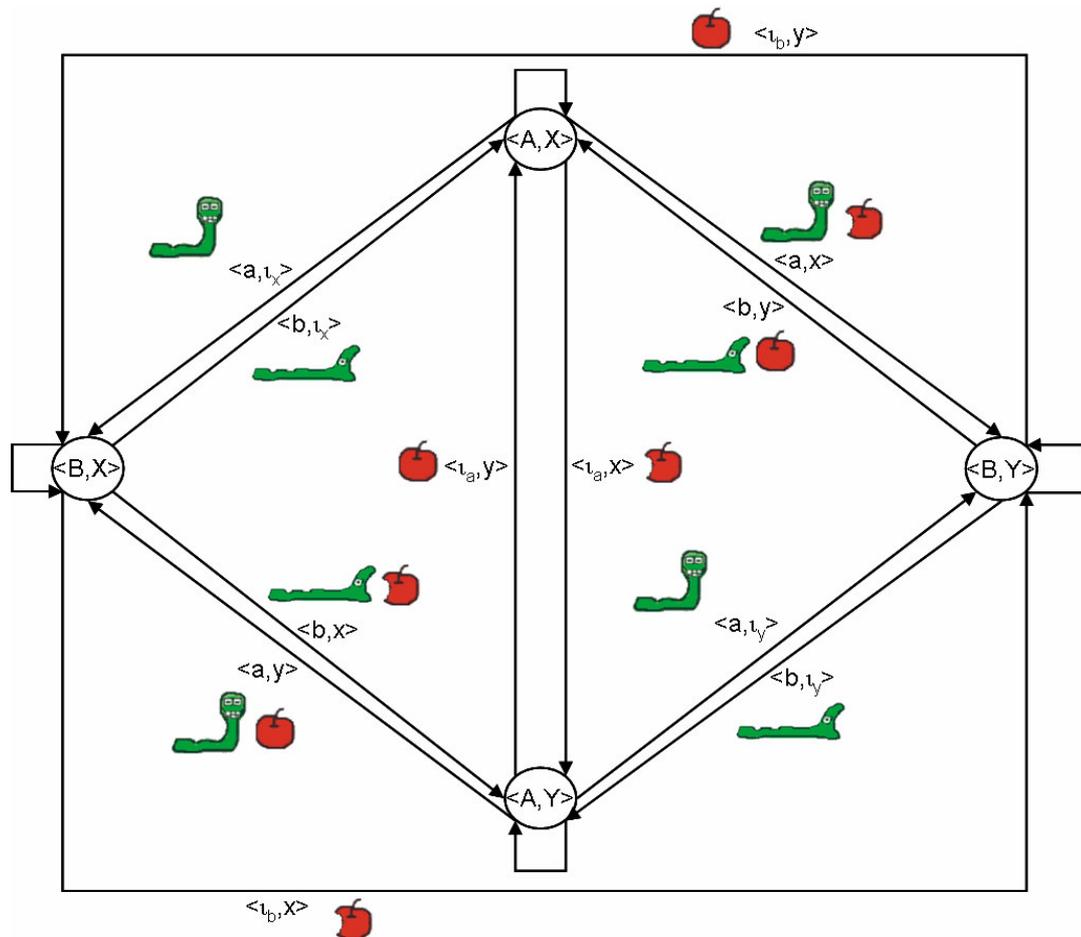


Figura 33 – Produto de grafos entre dois autômatos

O autômato resultante, Fig. 33, está desenhado como uma forma didática da explicação da operação do produto, que é obtido a partir da aplicação matemática da operação categorial de produto sobre os autômatos originais. Os autômatos originais são os únicos que precisam ser definidos, e esses continuam existindo e podem ser retomados a qualquer momento. Portanto, qualquer modificação feita sobre os autômatos originais será refletida automaticamente no resultado da operação do produto.

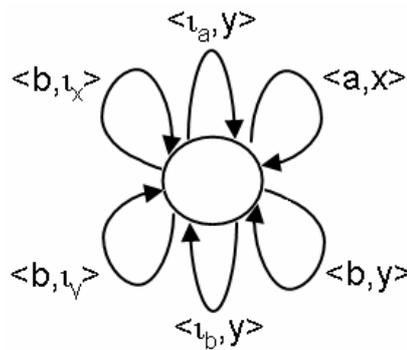
Conforme mencionado no capítulo anterior, uma das interpretações de produto fibrado em grafos é restringir um sistema, de forma a satisfazer determinadas condições, ou seja, uma parte do sistema original é restringida e a outra mantida preservando suas características.

O conceito de produto fibrado foi apresentado neste capítulo com ênfase em grafos. Sua aplicação entre autômatos, baseado em autômato finito, resulta em um novo autômato criado a partir dos autômatos originais. É importante ressaltar que o

que for modificado nos autômatos originais será modificado automaticamente no resultado do produto fibrado.

A operação categorial de produto fibrado poderá ser aplicada sobre qualquer operação categorial, até mesmo sobre autômatos originais. Assim, tal operação categorial será aplicada sobre o produto, representado pela Fig. 33, que simula animações computacionais baseadas em autômatos finitos, sejam eles determinísticos ou não determinísticos.

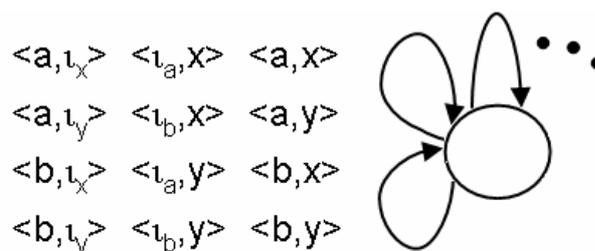
Suponhamos que seja desejado sincronizar as imagens do *Bicho* sorrindo (a) com a *Maçã* mordida (x), isso significa dizer que eles devem ocorrer simultaneamente, e os outros, b e y , podem ocorrer livremente. Na Fig. 34 é evidenciada tal restrição. Deste modo, é formada uma tabela de sincronizações (restrições) com os seguintes elementos:



Observação:

- O símbolo ι_k , em que $k \in \{A, B, X, Y\}$, representa a reflexão no estado k .

A tabela das sincronizações acima é parte da tabela com todas as arestas da composição paralela da Fig. 33, como é representada abaixo:



Portanto, para se obter a sincronização desejada, é necessário a partir do produto entre os autômatos excluirmos-se as transições que não fazem parte das restrições adotadas, ou seja, manter a e x e “apagar” as demais transições.

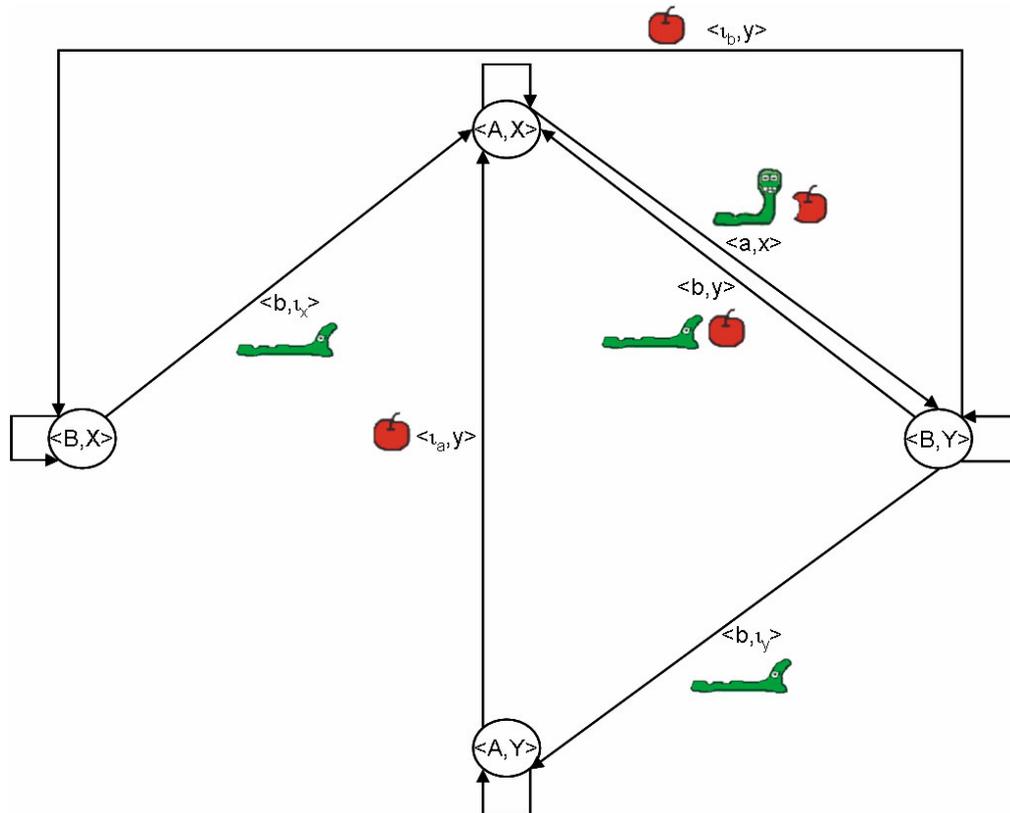


Figura 34 – Resultado da aplicação das restrições ao produto

Seguindo a idéia da restrição introduzida em **Set**, é necessário especificar um homomorfismo de grafos e uma inclusão no codomínio desse morfismo. A solução é ilustrada na Fig. 35:

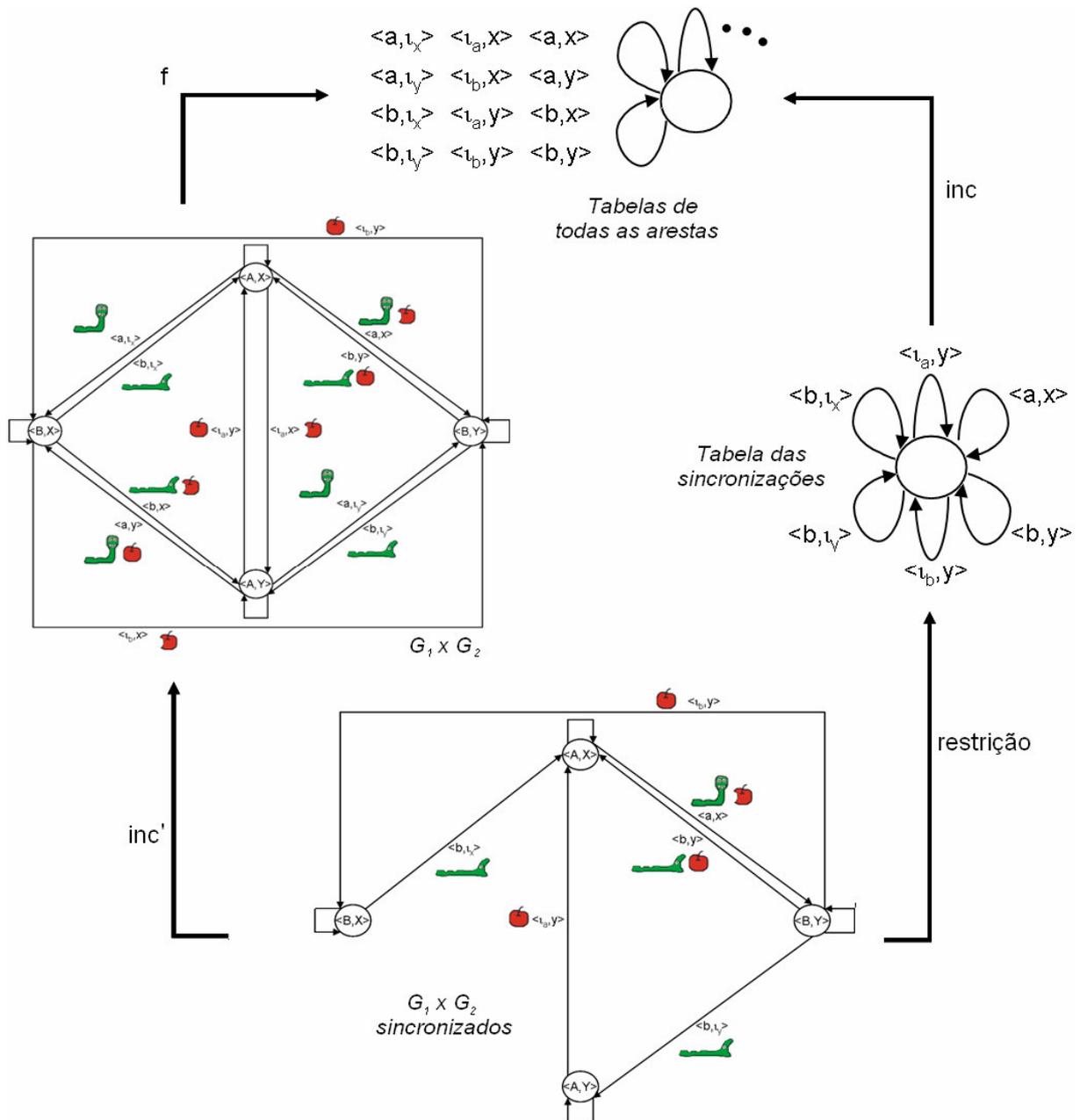


Figura 35 – Aplicação do produto fibrado baseado na restrição em Set

Onde:

As tabelas são representadas por grafos de um só vértice (MENEZES, 1996).

O morfismo *inc* é simplesmente uma inclusão da tabela de sincronizações na tabela de todas as arestas.

O morfismo *f* preserva todas as arestas do autômato produto ($G_1 \times G_2$).

O produto fibrado resulta em um autômato em que aparecem somente as arestas que se obteve da tabela de sincronizações, e dois morfismos, um de inclusão de grafos (*inc*) e outro que reflete a restrição de *f* para o codomínio desejado (*restrição*).

Assim, uma sincronização de sistemas pode ser expressa como uma restrição da composição paralela (produto categorial) dos sistemas componentes.

Em Teoria das Categorias é possível aplicar operações categoriais sobre operações categoriais, sejam elas quais forem e quantas vezes se ache necessário para que se obtenha o resultado desejado.

Fazendo uma comparação do resultado do sincronismo obtido através da operação categorial do produto fibrado com o modelo AGA já apresentado na seção 2.5, o mesmo resultado do produto fibrado como restrição de sistemas pode ser alcançado somente se a fita de entrada de cada um dos atores for alterada. Portanto, se tiver, por exemplo, cinco atores envolvidos na animação a fita de cada um deles deverá ser alterada para que a restrição desejada seja conseguida. Ou o autômato dos atores precisam ser alterados, ou se o projetista da animação desejar, um novo autômato deverá ser construído para que esse consiga a animação desejada.

Já com a utilização da operação categorial de produto fibrado demonstrado nesta seção, nem as fitas e nem os atores originais precisam ser alterados, a animação é obtida utilizando-se a fita já definida no modelo AGA e aplicando-se a operação sobre os autômatos dos atores originais. Como os autômatos originais não são modificados é possível retornar a utilizá-los quando se desejar.

Portanto, o auxílio das operações matemáticas categoriais em Teoria das categorias aplicado aos atores do modelo AGA tem uma grande vantagem, pois não precisamos ter a necessidade gerar um novo ator (autômato resultante) a partir dos atores originais para obter novas animações, basta-se apenas trabalhar com os autômatos originais, aplicando sobre eles as operações categoriais, podendo até mesmo, utilizar um ator quantas vezes for necessário. Isso diminui significativamente a necessidade de redefinir um ator sempre que se deseja utiliza-lo, obtendo dessa forma, descrições formais mais claras e simples.

5.2 Sincronização de Autômatos Temporizados

A utilização de autômatos temporizados proporciona a especificação do comportamento dinâmico entre os sistemas de transição e provê mecanismos de sincronização. Tal sincronização será abordada categorialmente nesta seção, conforme a definição do modelo de Alur e Dill, e com base nisso iremos demonstrar

uma outra forma de sincronização a partir da aplicação de operações categoriais para restringir sistemas de transição temporizados, proporcionando algumas vantagens referentes ao modelo AGA e ao próprio modelo de autômatos temporizados, o qual é estendido para o modelo AGA-S.

Conforme o modelo de Autômatos Temporizados proposto no capítulo 3, um autômato temporizado tem o objetivo de definir o comportamento de um sistema de transição de tempo real. Com base nisso, pretendemos delinear uma abordagem categorial para autômatos temporizados, modelando categorialmente a sua composição paralela – principalmente com o apoio dos estudos de grafos e da operação categorial de produto – e demonstrando como é possível sincronizá-los.

Tomando-se o conceito de autômato temporizado pela definição 12 e a característica da propriedade de composição descrita na seção 3.1, criamos uma extensão do modelo definido para o conceito de categoria, como segue:

Definição 25 *Um autômato temporizado G_{Atemp} é uma seis-upla:*

$$G_{Atemp} = \langle V, T, I, R, X, \partial_0, \partial_1 \rangle$$

Onde:

V – um conjunto de nodos ou vértices;

T – um conjunto de arcos ou arestas;

I - mapeamento que associa cada estado s em V com alguma restrição de relógio em $\Phi(X)$, ou seja, uma restrição invariável;

R - mapeamento que associa cada transição t em T com alguma restrição φ de relógio em $\Phi(X)$. A restrição de relógio φ determina quando a transição está habilitada;

X - conjunto finito de relógios;

$\partial_0, \partial_1: T \rightarrow V$ – são operações totais (funções) denominadas origem e destino, respectivamente. Um arco normalmente é denotado por $t: A \rightarrow B$, tal que $\partial_0(t) = A$ e $\partial_1(t) = B$.

As componentes I , R e X , foram estabelecidas para designar restrições aos estados e arestas conforme o modelo de autômatos temporizados estudado anteriormente. Deste modo ao realizarmos o produto entre dois autômatos temporizados (G_{Atemp1} e G_{Atemp2}) conforme definimos acima, temos que:

Definição 26 Sejam $G_{Atemp1} = \langle V_1, T_1, I_1, R_1, X_1, \partial_{01}, \partial_{11} \rangle$ e $G_{Atemp2} = \langle V_2, T_2, I_2, R_2, X_2, \partial_{02}, \partial_{12} \rangle$ autômatos (grafos) temporizados.

O produto dos autômatos é composto por:

1) O grafo:

$$G_{Atemp1} \times G_{Atemp2} = \langle V_1 \times V_2, T_1 \times T_2, I, R, X, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle$$

Onde:

$V_1 \times V_2$ – combinação de todos os vértices de G_{Atemp1} com G_{Atemp2} ;

$T_1 \times T_2$ – combinação de todas as arestas de G_{Atemp1} com G_{Atemp2} ;

I – conjunto de restrições invariáveis obtido pela conjunção das restrições dos autômatos originais;

R – conjunto de restrições sobre as arestas obtidas pela conjunção das restrições dos autômatos originais;

X – conjunto finito de variáveis de relógios de $X_1 \cup X_2$ e assumindo que $X_1 \cap X_2 = \emptyset$;

$\partial_{k1} \times \partial_{k2}$ – para $k \in \{0, 1\}$, são induzidos pelo produto em **Set** dos vértices e arestas como ilustrado na Fig. 36.

2) juntamente com os morfismos:

$$\pi_1 = \langle \pi_{V1}, \pi_{T1} \rangle: G_{Atemp1} \times G_{Atemp2} \rightarrow G_{Atemp1}$$

$$\pi_2 = \langle \pi_{V2}, \pi_{T2} \rangle: G_{Atemp1} \times G_{Atemp2} \rightarrow G_{Atemp2}$$

$$q_1 = \langle q_{R1}, q_{I1} \rangle: G_{Atemp1} \times G_{Atemp2} \rightarrow G_{Atemp1}$$

$$q_2 = \langle q_{R2}, q_{I2} \rangle: G_{Atemp1} \times G_{Atemp2} \rightarrow G_{Atemp2}$$

Onde π_1, π_2, q_1, q_2 são ilustrados na Fig. 36. As funções $t_{Rk1} \times t_{Rk2}$ e $t_{Ik2} \times t_{Ik1}$ mapeiam as restrições de relógios R nas arestas $T_1 \times T_2$ e as restrições invariáveis I nos vértices $V_1 \times V_2$.

Exemplo 8 De acordo com a definição 26, destacamos as exemplificações a seguir:

O conjunto de restrições invariáveis associados aos estados (l) da composição de autômatos temporizados seria, por exemplo, $l(v_1, v_2) = I_1(v_1) \wedge I_2(v_2) = x \leq 5 \wedge y \leq 5$.

O conjunto de restrições sobre as arestas (R) seria a conjunção das restrições das arestas dos autômatos originais, por exemplo, $R(t_1, t_2) = R_1(t_1) \wedge R_2(t_2) = z:=0 \wedge w \leq 1$.

O conjunto finito de relógios (X) agrega a união dos relógios ($X_1 \cup X_2$) dos autômatos componentes, por exemplo, $X_1 = \{x\}$ e $X_2 = \{y\}$ então $X = X_1 \cup X_2 = \{x\} \cup \{y\} = \{x, y\}$.

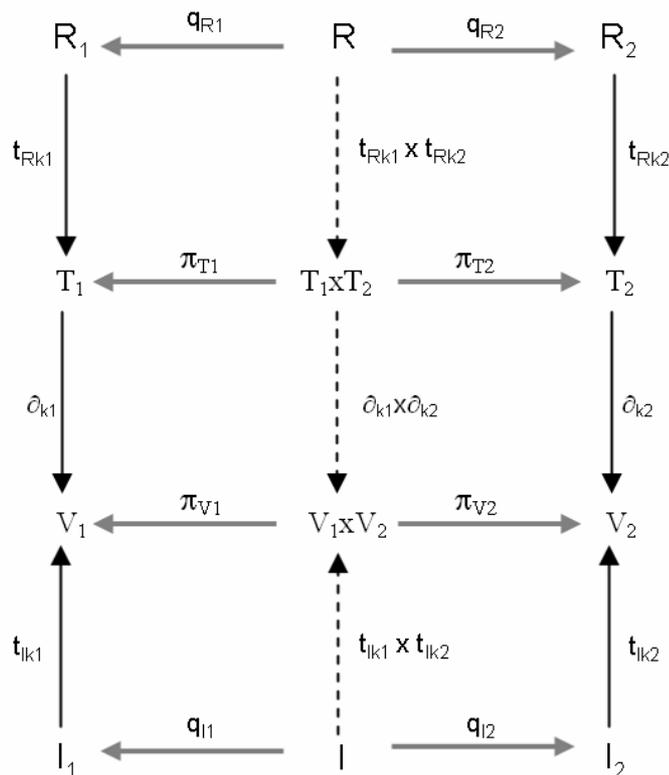


Figura 36 – Diagrama comutativo para produto de autômatos temporizados

Apesar da definição 25 não apresentar o estado inicial e os rótulos da categoria de um autômato temporizado, nos exemplos desta seção, estamos considerando tais exceções para representar sistemas de acordo com os exemplos do modelo de Alur e Dill estudados anteriormente no capítulo 3.

Considerando um sistema de transição S_{Atemp} , definido no capítulo 3, com a definição do produto categorial de autômatos temporizados, mostrado no início desta seção, já é possível determinar o sincronismo entre os autômatos componentes a partir dos símbolos comuns do alfabeto de entrada, de acordo com a composição de autômatos temporizados de Alur e Dill. A ocorrência de um símbolo comum determina que as transições correspondentes nesses autômatos devam ocorrer de forma

sincronizada. Como a saída está vinculada à transição dos autômatos temporizados, a sincronização garante que dois ou mais autômatos iniciem a apresentação dos objetos de saída ao mesmo tempo. Um exemplo do processo de sincronismo de autômatos temporizados é fornecido a seguir:

Tomando-se como base dois sistemas de transição temporizados, pode-se aplicar a operação categorial de produto para tais sistemas. O estado inicial dos sistemas de transição a serem utilizados está sendo apontado por uma seta. A Fig.37 representa os dois sistemas de transição temporizados, G_1 simbolizando o sistema de transição *Bicho* e G_2 simbolizando o sistema de transição *Maçã*, que compõem uma suposta animação.

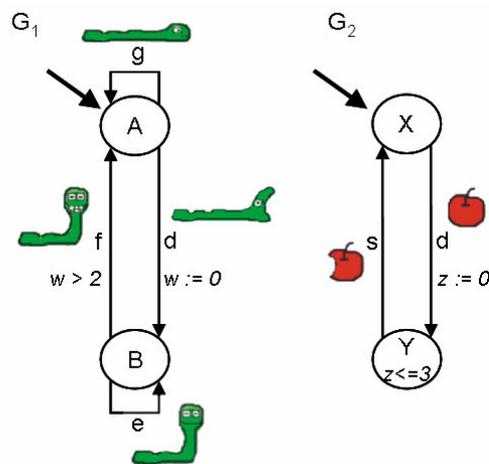


Figura 37 – Sistemas de transição temporizados de uma animação

Suponha que desejamos sincronizar as imagens do *Bicho* de boca aberta com a *Maçã* inteira, isso significa dizer que eles devem ocorrer simultaneamente, e os outros, podem ocorrer livremente. Para isso, toda transição de G_2 , que projeta a maçã inteira, é sincronizada com a transição de G_1 , que projeta o bicho com a boca aberta, utilizando um símbolo do alfabeto de entrada comum (d). O objeto resultante do produto para tal restrição, na Fig. 38, tem as transições como pares de transições dos grafos componentes, como na categoria de grafos. As saídas dos sistemas são conservadas nas novas transições de acordo com os estados de origem e destino dos sistemas componentes. Nota-se, portanto, que o produto resultante reflete a sincronização dos sistemas de transição temporizados componentes, respeitando suas propriedades originais.

É importante lembrar que é possível que o sistema resultante do produto possa produzir estados inalcançáveis tanto pela inexistência de caminhos quanto pelo aparecimento de limites impossíveis de serem satisfeitos, causados pela conjunção das restrições temporais. Desta maneira, estes estados podem ser eliminados sem prejuízo à composição dos sistemas de transição.

O sistema resultante, Fig. 38, está desenhado como uma forma didática da explicação da operação do produto, que é obtido a partir da aplicação matemática da operação categorial de produto sobre os sistemas de transição temporizados componentes, onde proporcionará a simulação das animações, sejam eles determinísticos ou não determinísticos. Os sistemas originais são os únicos que precisam ser definidos, e esses continuam existindo e podem ser retomados a qualquer momento. Portanto, qualquer modificação feita sobre os sistemas originais será refletida automaticamente no resultado da operação do produto.

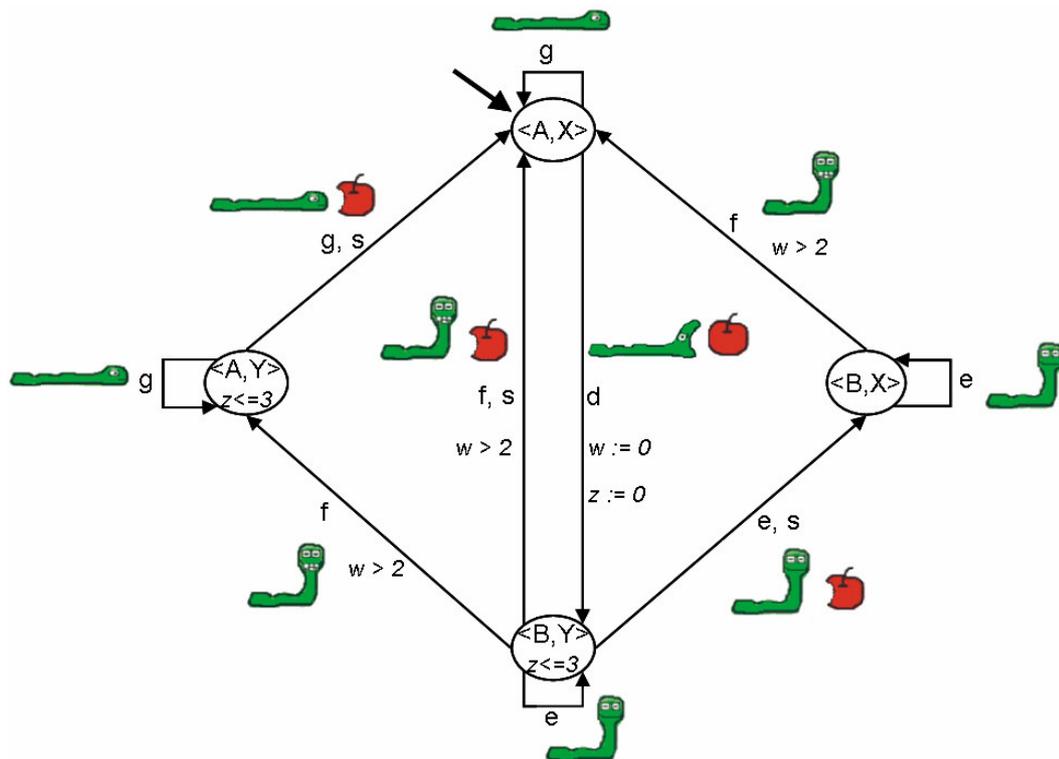


Figura 38 – Resultado do produto entre os sistemas com a restrição estabelecida

O produto categorial de autômatos temporizados, nesse caso, resulta em um sistema de transição temporizado baseado na restrição desejada, juntamente com as combinações dos demais estados, suas restrições invariáveis (se houverem), arestas e suas restrições (se houverem).

Assim, uma sincronização de sistemas de transição temporizados pode ser expressa como uma restrição dos sistemas componentes, ou seja, através dos símbolos comuns do alfabeto de entrada desses sistemas.

Uma outra forma de sincronizarmos sistemas de transição temporizados, não dependendo do alfabeto de entrada dos sistemas componentes – conforme a modelagem de Alur e Dill – e, com isso, obtermos uma vantagem sobre o modelo AGA-S, é realizarmos as operações categoriais de produto entre os sistemas de transição temporizados e em seguida o produto fibrado para restringirmos o sistema resultante de acordo com a sincronização desejada.

Tomando-se como base os dois sistemas de transição temporizados representados pela Fig.37, eliminamos o símbolo d de um dos autômatos que significaria a restrição associada ao outro autômatos. Deste modo, representaremos os dois autômatos de forma independente, como ilustrado na Fig. 39, para efetuarmos sua composição logo adiante.

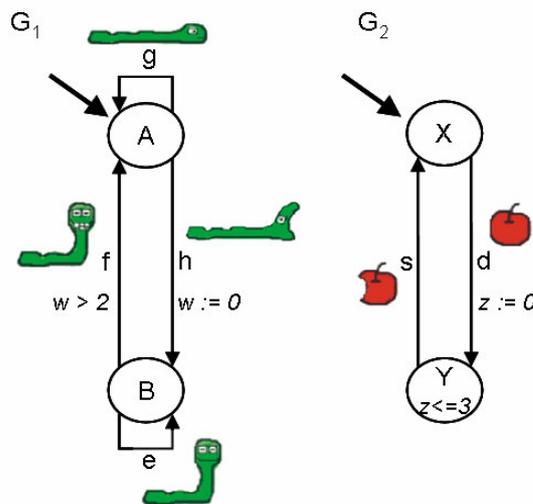


Figura 39 – Sistemas de transição temporizados independentes

A Fig. 40 ilustra o produto aplicado aos dois sistemas da Fig. 39, em que as transições são pares de transições juntamente com as restrições (se houverem) dos grafos componentes. A Fig. 40 apresenta a operação do produto entre os sistemas de transição, que é obtido a partir da aplicação matemática da operação categorial de produto sobre os sistemas originais. Os sistemas originais são os únicos que precisam ser definidos, e esses continuam existindo e podem ser retomados a qualquer

momento. Portanto, qualquer modificação feita sobre os autômatos originais será refletida automaticamente no resultado da operação do produto.

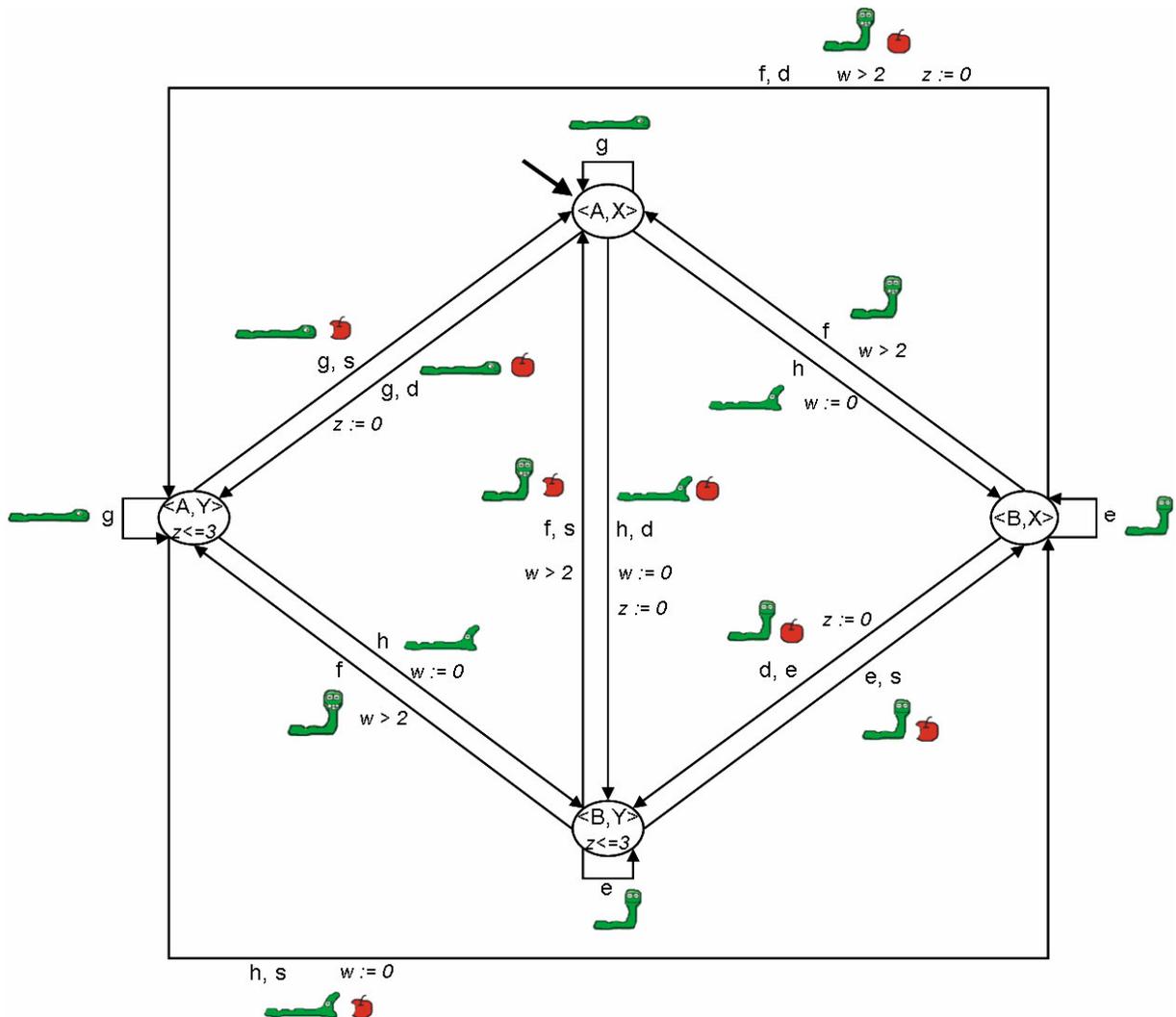


Figura 40 – Produto entre os dois sistemas de transição temporizados

É importante levarmos em consideração as restrições associadas às arestas e aos vértices da composição paralela entre os sistemas, lembrando, também, que é possível que o sistema resultante possa produzir estados inalcançáveis, tanto pela inexistência de caminhos quanto pelo aparecimento de limites impossíveis de serem satisfeitos causados pela conjunção das restrições temporais. Sendo assim, estes estados podem ser eliminados sem prejuízo à composição dos sistemas de transição.

A partir do produto entre os sistemas, apresentado pela Fig. 40, podemos então sincronizar os elementos de saída que desejamos através da aplicação matemática categorial do produto fibrado, sem precisarmos identificar os mesmos símbolos de entrada dos sistemas componentes para tal sincronização.

De acordo com a mesma idéia da restrição em **Set** apresentado na seção 4.8.1 e a aplicação do produto fibrado em autômatos não temporizados, podemos restringir de forma semelhante o sistema resultante da composição dos sistemas componentes.

Suponha que desejamos sincronizar as imagens do *Bicho* sorrindo mostrando seus dentes (f) com a *Maçã* mordida (s), isso significa dizer que eles devem ocorrer simultaneamente, e as outras, podem ocorrer livremente. Na Fig. 41 é evidenciada tal restrição, desta maneira é formada uma tabela de sincronizações (restrições) com o alfabeto de entrada associando seus objetos de saída e suas respectivas restrições temporais, tais como: $\langle f, s \rangle$ com a restrição $w > 2$, $\langle g \rangle$, $\langle h \rangle$ com a inicialização de relógio $w := 0$, $\langle e \rangle$, $\langle d, e \rangle$ com a inicialização de relógio $z := 0$, $\langle g, d \rangle$ com a inicialização de relógio $z := 0$ e $\langle h, d \rangle$ com as inicializações dos relógios $z := 0$ e $w := 0$.

A tabela das sincronizações é parte da tabela com todas as arestas da composição paralela da Fig. 40: $\langle f \rangle$ com a restrição $w > 2$, $\langle e, s \rangle$, $\langle f \rangle$ com a restrição $w > 2$, $\langle g, s \rangle$, $\langle f, d \rangle$ com as restrições $w > 2$ e $z := 0$, $\langle h, s \rangle$ com a inicialização de $w := 0$, $\langle f, s \rangle$ com a restrição $w > 2$, $\langle g \rangle$, $\langle h \rangle$ com a inicialização de relógio $w := 0$, $\langle e \rangle$, $\langle d, e \rangle$ com a inicialização de relógio $z := 0$, $\langle g, d \rangle$ com a inicialização de relógio $z := 0$ e $\langle h, d \rangle$ com as inicializações dos relógios $z := 0$ e $w := 0$.

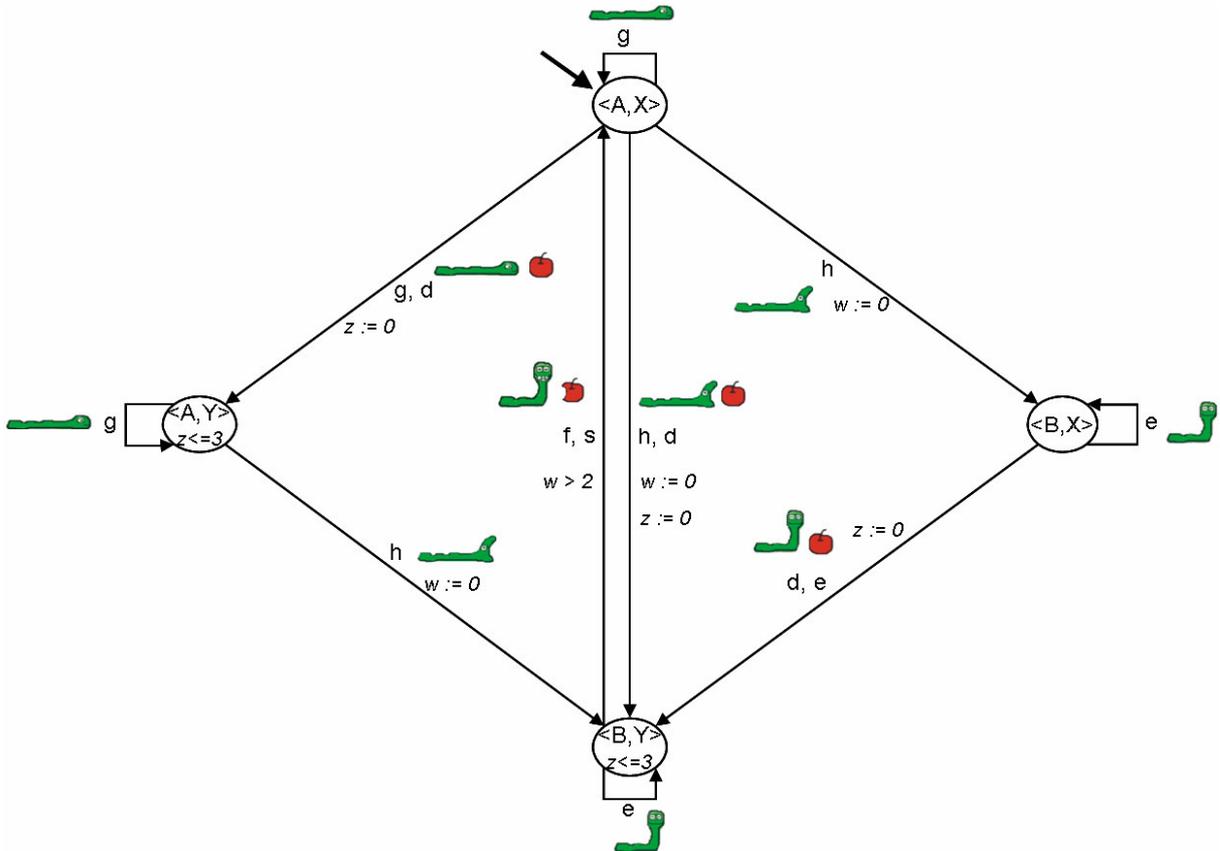


Figura 41 – Resultado da restrição ao produto entre os autômatos temporizados

Seguindo a idéia da restrição de autômatos não temporizados, de acordo com a seção 5.1, é necessário especificar um homomorfismo de grafos e uma inclusão no codomínio desse morfismo. A solução é ilustrada na Fig. 42:

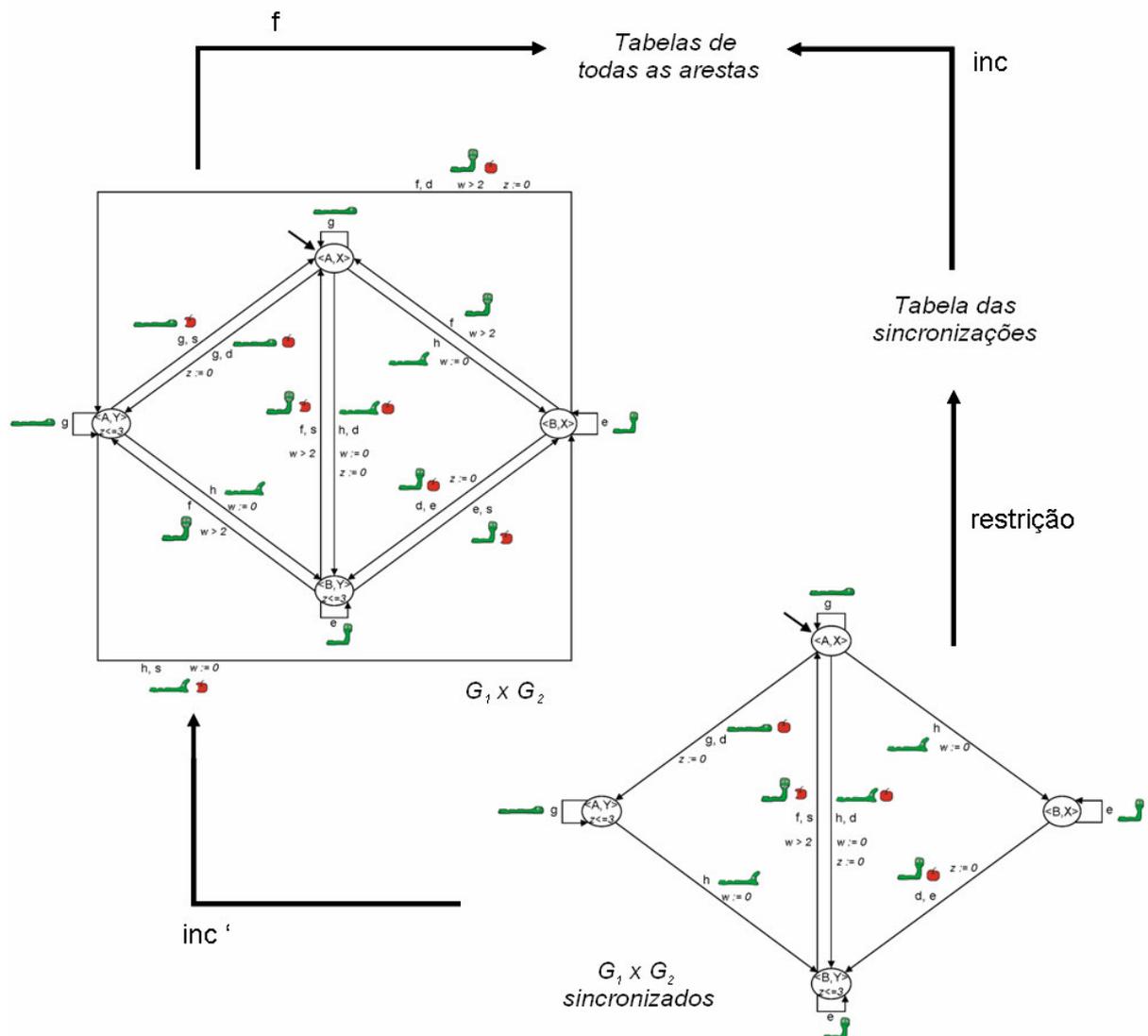


Figura 42 – Aplicação do produto fibrado na composição paralela dos sistemas componentes

Conforme a Fig. 42: as tabelas representam o mapeamento dos símbolos de entradas com as restrições a serem satisfeitas e suas respectivas imagens de saída na transição entre um estado e outro. O morfismo *inc* é simplesmente uma inclusão da tabela de sincronizações na tabela de todas as arestas. O morfismo *f* preserva todas as arestas do autômato produto ($G_1 \times G_2$).

O produto fibrado resulta em um sistema em que aparecem somente as arestas que se obteve da tabela de sincronizações, e dois morfismos, um de inclusão de grafos temporizados (*inc'*) e outro que reflete a restrição de *f* para o codomínio desejado (*restrição*).

Com isso, uma sincronização de sistemas pode ser expressa como uma restrição da composição paralela dos sistemas componentes. A restrição desejada

ainda determina outras sincronizações, por exemplo, que o *Bicho* esteja esticado durante 3 segundos no estado $\langle A, Y \rangle$ para que depois ocorra a transição e ele aparecer esticado com a boca aberta.

Mais uma vez é notável a aplicação de operações categoriais sobre operações categoriais, sejam elas quais forem e quantas vezes se ache necessário para que se obtenha o resultado desejado.

Este processo de sincronização de sistemas de transição temporizados proporcionou uma grande vantagem ao se comparar com o tipo de sincronismo dos atores no modelo AGA-S, que é uma extensão do modelo AGA, pois deixamos independentes as fitas e os sistemas originais, ou seja, eles não precisam ser alterados. A animação é obtida utilizando a mesma fita definida pelo modelo e somente é feita aplicação das operações categoriais, produto e produto fibrado, para resultar na sincronização que se deseja sobre os sistemas originais. E, como os sistemas originais não são modificados, é possível retornar a utilizá-los quando se desejar.

6 Conclusões e Trabalhos Futuros

Este trabalho apresentou a Teoria das Categorias aplicada à sincronização de autômatos e o desenvolvimento de uma nova abordagem categorial, utilizando o produto fibrado, uma das ferramentas mais poderosas em Teoria das Categorias. Tal abordagem foi desenvolvida com o objetivo de tratar tanto da sincronização de autômatos não temporizados, quanto da sincronização de autômatos temporizados, através da utilização do produto categorial e do produto fibrado baseado em grafos, onde o grande diferencial desses dois tipos de sincronização está na utilização de restrições temporais dos relógios, que definem o comportamento dinâmico quanto ao tempo de permanência nos estados e os instantes disponíveis para transições, presentes na sincronização dos autômatos temporizados.

O capítulo 5 apresenta toda a investigação sobre a aplicação de Teoria das Categorias para realizar tanto a sincronização de autômatos não temporizados quanto a sincronização de autômatos temporizados, sendo demonstrados através de seus respectivos exemplos, que compõem uma animação computacional com as restrições desejadas. Para se chegar a esta animação, em um primeiro momento se pensa que é necessário que se crie um novo autômato, e que provavelmente este novo autômato geraria uma explosão de estados e arestas. Porém nenhum novo autômato é criado e a animação pode ser gerada, baseada fundamentalmente nas operações matemáticas categoriais já investigadas. As animações resultantes serviram para verificar o resultado prático da aplicação dessas operações.

É possível observar que as operações categoriais são poderosas o suficiente para tratar a composição de sistemas, as restrições que se querem através do produto fibrado, e ainda aplicar operações categoriais sobre operações categoriais (produto fibrado sobre o produto, por exemplo) quantas vezes se ache necessário para que se obtenha o resultado desejado. Tal situação considerada complexa, de difícil resolução pôde ser solucionada de forma simples e clara, utilizando a Teoria das Categorias.

Além disso, uma das vantagens dessa abordagem, através do resultado das operações categoriais utilizadas, é que os autômatos originais não são perdidos, podendo assim serem reutilizados quantas vezes forem necessários. A noção de

reuso é uma das características marcantes em Teoria das Categorias.

A principal contribuição desta monografia está na abordagem categorial da sincronização de autômatos temporizados. O modelo estudado de autômatos temporizados proposto por Alur e Dill, no capítulo 3, representa a composição de autômatos temporizados como determinação do sincronismo a partir dos símbolos comuns do alfabeto de entrada, em que esta operação possibilita que sistemas complexos possam ser definidos a partir do produto dos sistemas componentes. Tal modelo foi abordado categorialmente e exemplificado através da composição de sistemas de transição que formavam uma animação.

A partir deste resultado, realizamos uma nova abordagem categorial do sincronismo para os sistemas de transição temporizados, em que não dependemos das modificações dos símbolos de entrada (igualdade de símbolos) dos sistemas componentes para fornecer a sincronização desejada. A grande vantagem está na independência das fitas e dos sistemas originais, ou seja, eles não necessitam ser alterados, e mais ainda na aplicação das operações categorias, produto e produto fibrado, respectivamente, pois determinaram o sincronismo que se desejava sobre os sistemas originais. E como foi destacado anteriormente não há necessidade de criar outros autômatos para gerar novas animações, conforme foi ressaltado na explicação dos exemplos do capítulo 5, e como os sistemas originais não são modificados é possível retornar a utilizá-los quando se desejar.

Tal contribuição foi satisfatória por fornecer vantagens utilizando uma abordagem categorial para sincronização de autômatos, modelando os comportamentos de sistemas não temporizados e temporizados, baseados nos estudos de Teoria dos Autômatos, Autômatos Temporizados e Teoria das Categorias.

A abordagem de Teoria das Categorias nesta monografia irá contribuir como referencial para futuros trabalhos de pesquisa, e também para a consolidação do grupo de Teoria das Categorias na área de Computação Teórica, já que é um campo emergente na UFPel.

Embora o objetivo deste trabalho não fosse implementar as operações categoriais estudadas para demonstrar o sincronismo e sim interpretá-las, pois a motivação deste projeto é teórica, percebe-se que o trabalho proposto é motivado por uma aplicação prática.

Propomos como trabalhos futuros a implementação das operações categoriais, produto categorial e produto fibrado, para gerar animações de forma a

estender o modelo AGA, por exemplo, proporcionando desta maneira as grandes vantagens já citadas no decorrer da monografia.

7 Referências

ACCORSI, Fernando; MENEZES, Paulo Fernando Blauth. Animação Gráfica Baseada na Teoria de Autômatos. In: WORKSHOP DE MÉTODOS FORMAIS, 3., 2000, João Pessoa, **Anais do...** João Pessoa: SBC, 2000. p.122 – 127.

ACCORSI, Fernando; MENEZES, Paulo Fernando Blauth; NEDEL, Luciana Porcher. Animação Gráfica Baseada em Autômatos Temporizados Sincronizados. In: WORKSHOP DE MÉTODOS FORMAIS, 4., 2001, Rio de Janeiro. **Anais do...** Rio de Janeiro: SBC, 2001. p.75 – 80.

ACCORSI, Fernando. **Animação Bidimensional para World Wide Web Baseada em Autômatos Finitos**. 2002. 112f. Dissertação (Mestrado em Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre.

ALUR, Rajeev, DILL, David. A theory of timed automata. In: THEORETICAL COMPUTER SCIENCE, 1994, Essex, United Kingdom. **Proceedings of the Real-Time: Theory in Practice**, REX Workshop, v.126, p. 183-235, 1994.

ALUR, Rajeev, DILL, David. Automata-theoretic verification of real-time systems. In: TRENDS IN SOFTWARE SERIES, 1996. **Proceedings of the Formal Methods for Real-Time Computing**, p. 55-82, 1996.

ALUR, Rajeev. Timed automata. In: LECTURES NOTES IN COMPUTER SCIENCE, 1999, Trento, Itália. **Proceedings of the 11th International Conference on Computer Aided Verification**, v. 1633, p. 8-22, 1999.

ASPERTI, Andrea; LONGO, Giuseppe. **Categories Types and Structures: An Introduction to Category Theory for the working computer scientist**. MIT Press, 1991. 293p. Disponível em: <<ftp://ftp.di.ens.fr/pub/users/longo/CategTypesStructures/book.pdf>>.

AUTÔMATO. Disponível em: <<http://pt.wikipedia.org/wiki/Aut%C3%B4matos>>. Acesso em: 5 ago. 2006.

AUTÔMATOS. Disponível em: <<http://www.comp.ita.br/~marck/TrabAutom/Automato%20com%20Sa%EDda/index.html>>. Acesso em: 20 jul. 2006.

BRITO, Roberta C. de; MARTENDAL, Diogo M.; OLIVEIRA, Henrique Eduardo M. de. **Máquinas de estados finitos de Mealy e Moore**. Disponível em: <http://www.inf.ufsc.br/~barreto/trabaluno/TC_roberta_diogo_henrique.pdf#search=%22roberta%20diogo%20moore%20mealy%22>. Acesso em: 6 ago. 2006.

CAMPANI, Carlos Antônio Pereira. **Teoria das Categorias**. [133-151], 157 slides, 2005. Disponível em: <<http://minerva.ufpel.edu.br/~campani/cat.pdf>>.

CAMPANI, Carlos Antônio Pereira. **Disciplina de Linguagens Formais**. [10-100], 263 slides, 2006. Disponível em:
<<http://minerva.ufpel.edu.br/~campani/lingformal.pdf>>.

DIRETRIZES Curriculares de Cursos da Área de Computação e Informática. Disponível em: <<http://www.pp.ufu.br/Computacao.htm>>. Acesso em: 28 out. 2005.

DIESTEL, Reinhard. **Graph Theory**. Springer-Verlag, v. 173 of Graduate Texts in Mathematics, 2005. 410p. Disponível em:
<<http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/GraphTheoryIII.pdf>>.

FEOFILOFF, Paulo; KOHAYAKAWA, Yoshiharu; WAKABAYASHI, Yoshiko. **Uma Introdução sucinta à Teoria de Grafos**. II Bienal da SBM, 2005, 61p. Disponível em:
<<http://www.ime.usp.br/~pf/teoriadosgrafos/>>.

HAEUSLER, Edward Herman; MENEZES, Paulo Fernando Blauth. **Teoria das Categorias para Ciência da Computação**. 1ed. Porto Alegre, Brasil: Sagra Luzzatto, 2001. 314p.

HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. E. **Introduction to Automata Theory, Languages and Computation**. 2ed. Addison Wesley, 2001. 521 p.

HOPCROFT, J. E.; ULLMAN, J. D. **Introduction to Automata Theory, Languages and Computation**. Addison-Wesley, 1979.

MAC LANE, Saunders. **Categories for the Working Mathematician**. 2ed. New York, USA: Springer, 1971. 302p.

MACHADO, Júlio Henrique de Araújo Pereira. **Hyper-Automaton: Hipertextos e Cursos na Web usando Autômatos Finitos com Saída**. 2000. 149p. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre.

MENEZES, Paulo Fernando Blauth. **Reificação de Objetos Concorrentes**. 1996. 148f. Tese (Doutorado em Matemática) – Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa.

MENEZES, Paulo Fernando Blauth. **Linguagens Formais e Autômatos**. 3ed. Porto Alegre, Brasil: Sagra Luzzatto, 1997. p. 41 – 86.

REIS, Carla Alessandra Lima. **Linguagens Formais**. Belém, Brasil. 2002. 57p. Disponível em:
<<http://www.cultura.ufpa.br/clima/lformais/apostila.pdf#search=%22apostila.pdf%20aut%C3%B4matos%22>>.

SAITO, Daniela Satomi. Visualização de Grafos Aplicada a Autômatos Finitos com saída. In: SEMANA ACADÊMICA DO PPGC DA UFRGS, 8., 2001, Porto Alegre. **Anais do...** Porto Alegre: UFRGS, 2001.