

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CHRISTINE VIEIRA SCARPATO

**Investigação e Aplicação de Operações
Categoriais entre Atores de Animações
Baseadas em Autômatos Finitos com Saída**

Dissertação submetida à avaliação, como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Dr. Paulo Fernando Blauth Menezes
Orientador

Dr.^a Luciana Porcher Nedel
Co-Orientadora

Porto Alegre, janeiro de 2004

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Scarpato, Christine Vieira

Investigação e Aplicação de Operações Categoriais entre Atores de Animações Baseadas em Autômatos Finitos com Saída / Christine Vieira Scarpato. – Porto Alegre: PPGC da UFRGS, 2003. 119 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2004. Orientador: Paulo Fernando Blauth Menezes. Co-Orientadora: Luciana Porcher Nedel

1. teoria dos autômatos 2. animação 3. Teoria das Categorias. I. Menezes, Paulo F. Blauth. II. Nedel, Luciana Porcher. III. Investigação e Aplicação de Operações Categoriais entre Atores de Animações Baseadas em Autômatos Finitos com Saída.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^ª. Wranna Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitora Adjunta de Pós-Graduação: Prof^ª. Jocélia Grazia

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Quando se encerra um trabalho, é difícil colocar em uma folha todas as pessoas que se deseja agradecer, pois são tantos os envolvidos e torcendo para que tudo de certo.

Tantas são as dificuldades enfrentadas, hora emocionalmente, hora fisicamente, porém todos que me cercam sempre deram forças para que eu não desistisse no meio do caminho.

Não posso deixar de agradecer meu grande companheiro, hoje meu marido Sandro Scarpato, afinal de contas sempre esteve ao meu lado e durante o mestrado me deu um filho maravilhoso, Arthur Vieira Scarpato, que hoje é a razão das nossas vidas. Ele, tanto quanto eu, lutou e batalhou para que este trabalho fosse concluído.

Devo agradecer também minha mãe, Neusa Maria Maier Vieira, sempre companheira das minhas idas a Porto Alegre. Esta também sempre me deu forças e me apoiou nos momentos de desânimo e angústia. Meu irmão, Alexandre Maier Vieira, que sempre esteve pronto a me ajudar.

Agradeço também ao meu orientador, Paulo Blauth Menezes, que acreditou em mim, mesmo quando eu não acreditava que podia concluir este trabalho. Nunca me deixou desistir, e da sua forma soube me conduzir até o final. Agradeço também a minha co-orientadora, Luciana Porcher Nedel.

Gostaria também de agradecer aos alunos do grupo de estudo do professor Blauth, foram tantos que me ajudaram de uma maneira ou de outra, alguns até já se desvincularam do grupo, que nem me atrevo a colocar os nomes com receio de esquecer de algum deles.

Não posso esquecer de Deus, que muitas vezes foi o único que conseguiu me escutar nos dias de nervosismo, tristeza e angústia. E hoje devo agradecer-lo por ter chego onde cheguei.

No fundo eu sabia que podia chegar até aqui, mas mesmo assim tive momentos de fraqueza, como qualquer ser humano. Mas hoje concluo esta dissertação com a certeza de ter crescido muito em termos de conhecimento intelectual e de conhecimento de vida.

*A vitória foi sempre, de quem nunca duvidou dela.
(Autor Desconhecido)*

SUMÁRIO

LISTA DE ABREVIATURAS.....	7
LISTA DE FIGURAS.....	8
RESUMO.....	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Motivação	13
1.2 Contextualização.....	13
1.3 Problemática	15
1.4 Objetivo	15
1.5 Descrição e Apresentação da Dissertação	16
2 PROJETO HYPER SEED.....	18
2.1 Introdução	18
2.2 Hyper-Automaton.....	18
2.3 Hyper-Automaton: Avaliação Interativa	19
2.4 eXtensible Hyper-Automaton (XHA)	20
2.5 Animação Bidimensional para World Wide Web (AGA).....	21
2.6 Aga Player	22
2.7 Astrha	24
2.8 Considerações Finais	24
3 ANIMAÇÕES POR COMPUTADOR	26
3.1 Introdução	26
3.2 Características da Web	28
3.3 Animações bidimensionais.....	29
3.3.1 Macromedia Flash (SWF)	29
3.3.2 Scalable Vector Graphics (SVG).....	30
3.3.3 Graphics Interchange Format (GIF)	32
3.3.4 Moving Picture Expert Group (MPEG-1), Quick Time e Audio Video Interleave (AVI)	33
3.4 Considerações Finais	34
4. MODELO AGA	35
4.1 Introdução	35
4.2 Autômatos Finitos com Saída	36
4.3 Função de Saída Contextual.....	36
4.4 Função de Transição Temporal	37
4.5 Animação como um Conjunto de Atores no modelo AGA	38
4.6 Considerações Finais	40

5 AGAND (ANIMAÇÃO GRÁFICA BASEADA EM AUTÔMATOS FINITOS NÃO DETERMINÍSTICOS)	41
5.1 Introdução	41
5.2 Visão Geral do modelo AGAND	41
5.3 Não determinismo interno e externo.....	42
5.4 Utilização de pesos nas arestas do AFND – AFNDp	42
5.5 Considerações Finais	43
6 APLICAÇÃO DE PRODUTO E COPRODUTO ENTRE ATORES DE UMA ANIMAÇÃO.....	45
6.1 Introdução	45
6.2 Definição de Produto e Coproduto em Grafos.....	45
6.3 Aplicação do Produto	46
6.3.1 Fita de entrada	48
6.4 Aplicação do Coproduto	49
6.4.1 Fita de entrada	50
6.4.2 Comparação com o modelo AGA	51
6.5 Produto e Coproduto de Grafos Reflexivos	52
6.5.1 Aplicação do Produto de Grafos Reflexivos	54
6.5.2 Aplicação do Coproduto de Grafos Reflexivos.....	55
6.6 Aplicação de Operações sobre Operações	56
6.7 Considerações Finais e Trabalhos Futuros	57
7 APLICAÇÃO DE PRODUTO FIBRADO E SOMA AMALGAMADA ENTRE ATORES DE UMA ANIMAÇÃO	58
7.1 Introdução	58
7.2 Definição de Produto Fibrado e Soma Amalgamada em Grafos	58
7.3 Produto Fibrado como restrição de sistema	60
7.3.1 Fita de entrada	62
7.3.2 Comparação com o AGA	63
7.4 Produto Fibrado como sincronizador de sistemas	63
7.4.1 Fita de entrada	64
7.5 Soma Amalgamada sem identificação das saídas	65
7.5.1 Fita de entrada	66
7.6 Soma Amalgamada com identificação das saídas.....	67
7.7 Operações sobre operações	69
7.8 Considerações Finais e Trabalhos Futuros	70
8 SUGESTÃO DE IMPLEMENTAÇÃO	72
8.1 Introdução	72
8.2 Visão Geral.....	72
8.3 Produto	74
8.4 Coproduto	75
8.5 Produto Fibrado como restrição de sistema	76
8.6 Produto Fibrado como sincronizador de sistema	77
8.7 Soma Amalgamada sem identificação das saídas	78
8.8 Soma Amalgamada com a identificação das saídas.....	79
8.9 Considerações Finais	80
9 ANIMAÇÃO COMPOSTA POR OPERAÇÕES SOBRE OPERAÇÕES	81
9.1 Introdução	81

9.2 Atores e Cena	81
9.3 Resolução.....	84
9.4 Considerações Finais.....	88
10 CONCLUSÃO E TRABALHOS FUTUROS.....	89
10.1 Conclusões.....	89
10.2 Trabalhos Futuros.....	91
10.3 Produção Científica.....	91
REFERÊNCIAS	93
ANEXO A PROJETO <i>HYPER SEED</i>	97
ANEXO B ARTIGO PUBLICADO NA REVISTA DE TECNOLOGIA E AMBIENTE DA UNESC	104
ANEXO C PROJETO PNEE-AUTÔMATO	113

LISTA DE ABREVIATURAS

AFD	Autômato finito determinístico
AFND	Autômato finito não determinístico
AFNDp	Autômato finito não determinístico com pesos
AFS	Autômato finito com saída
AGA	Animação gráfica baseada em autômatos finitos
AGAND	Animação gráfica baseada em autômatos finitos não determinísticos
API	Application Program Interface
AVI	Audio Video Interleave
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DTD	Document Type Declaration
DOM	Document Object Model
EAD	Educação a distância
ECMA	European Computer Manufacturers Association
GIF	Graphics Interchange Format
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JPEG	Joint Photographic Experts Group
JSP	Java Server Pages
ms	milésimos de segundos
MPEG	Moving Picture Expert Group
OO	Orientação a objetos
PPGC	Programa de Pós Graduação da Computação
RGB	Red, Green and Blue
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics
SWF	Shockwave Flash
UFRGS	Universidade Federal do Rio Grande do Sul
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

LISTA DE FIGURAS

Figura 1.1: Pesquisas baseadas em modelos de autômatos com saída	12
Figura 1.2: Construção de um curso	15
Figura 2.1: Painel de controle do AGA Player visualizado pelo observador	23
Figura 3.1: Processamento do SWF no programa de visualização	30
Figura 3.2: Animação de um retângulo em SVG	31
Figura 3.3: Estrutura do GIF versão 89 ^a	33
Figura 4.1: Atores especificados no modelo AGA.....	35
Figura 4.2: Troca de imagens do ator bicho provocadas pela leitura da fita	37
Figura 4.3: Animação envolvendo dois atores AGA.....	39
Figura 5.1: Animação modelada no AGANd	42
Figura 5.2: Utilização de um AFNDp no modelo AGANd.....	43
Figura 6.1: Produto de Grafos	46
Figura 6.2: Coproduto de Grafos	46
Figura 6.3: Atores de uma animação	47
Figura 6.4: Produto entre atores de uma animação	48
Figura 6.5: Fita de entrada única para o Produto entre atores de uma animação	49
Figura 6.6: Coproduto entre atores de uma animação	50
Figura 6.7: Fita de entrada para o Coproduto	51
Figura 6.8: Atores do modelo AGA	51
Figura 6.9: Animação	52
Figura 6.10: Produto de Grafos Reflexivos	53
Figura 6.11: Coproduto de Grafos Reflexivos	53
Figura 6.12: Diagrama comutativo adicional para produto de grafos reflexivos	53
Figura 6.13: Diagrama comutativo adicional para coproduto de grafos reflexivos	54
Figura 6.14: Atores de uma animação	54
Figura 6.15 - Produto de grafos reflexivos entre dois atores.....	55
Figura 6.16: Coproduto de grafos reflexivos entre dois atores	56
Figura 6.17: Atores de uma animação	56
Figura 6.18: Produto entre o Ator 1 e o resultado do Coproduto entre os Atores 2 e 3	57
Figura 7.1: Produto Fibrado de grafos como operador do tipo restrição.....	59
Figura 7.2: Soma Amalgamada de grafos como operador de sistemas	60
Figura 7.3: Atores de uma animação	61
Figura 7.4: Produto entre dois atores de uma animação.....	61
Figura 7.5: Resultado do Produto Fibrado entre dois atores de uma animação	62
Figura 7.6: Animação do Produto Fibrado como restrição de sistema.....	63

Figura 7.7: Produto Fibrado como sincronizador de sistema	64
Figura 7.8: Animação de Produto Fibrado como sincronizador de sistema.....	65
Figura 7.9: Atores de uma animação	65
Figura 7.10: Soma Amalgamada entre dois atores de uma animação	66
Figura 7.11: Animação da Soma Amalgamada envolvendo dois atores	67
Figura 7.12: Atores de uma animação	67
Figura 7.13: Soma Amalgamada identificando as saídas	68
Figura 7.14: Animação da Soma Amalgamada envolvendo dois atores	69
Figura 7.15: Atores de uma animação	69
Figura 7.16: Soma Amalgamada sem identificação das saídas entre ator1 e ator2.....	70
Figura 7.17: (Soma Amalgamada, sem identificação das saídas entre ator1 e ator2) x ator3	70
Figura 8.1: Ilustração da implementação.....	73
Figura 8.2: Mapeamento das fitas de entrada	74
Figura 8.3: Animação obtida pela leitura da fita	75
Figura 8.4: Animação obtida pela leitura da fita	76
Figura 8.5: Animação obtida pela leitura da fita	77
Figura 8.6: Animação obtida pela leitura da fita	78
Figura 8.7: Animação obtida pela leitura da fita	79
Figura 8.8: Animação obtida pela leitura da fita	80
Figura 9.1: P (Pessoa caminhando)	82
Figura 9.2: M (Pessoa que estende a mão para cumprimentar).....	82
Figura 9.3: Chuta (Pessoa chutando bola)	83
Figura 9.4: Goleiro	83
Figura 9.5: Carro	83
Figura 9.6: Sol	83
Figura 9.7: Dia ensolarado e Dia nublado	85
Figura 9.8: Avião e Pássaros Voando.....	86
Figura 9.9: Pedaco da cena.....	88

RESUMO

O presente trabalho apresenta uma investigação sobre algumas operações categoriais baseadas em grafos e a aplicação das mesmas a uma área específica da Ciência da Computação, a saber, animações computacionais baseadas em autômatos finitos com saída.

As operações categoriais estudadas neste trabalho são: Produto, Coproduto, Soma Amalgamada e Produto Fibrado. O modelo AGA (Animação Gráfica baseada em Autômatos finitos) foi o escolhido para ser utilizado como base desta dissertação.

Inspirado nestes estudos, o trabalho contém uma proposta de como aplicar tais operações com o objetivo de definir animações aparentemente complexas, de forma simples, precisa e de fácil implementação. O enfoque está baseado em J. Stoy que diz que um dos objetivos para o uso da semântica formal de teoria das categorias é “sugerir meios ou formas para o projetista desenvolver sistemas melhores, mais elegantes (“limpos”) e com descrições formais mais simples”. Entretanto, não é objetivo deste trabalho verificar se a utilização destas operações é ou não melhor do que a utilização de qualquer outra solução para criar novas animações.

Esta dissertação traz uma nova versão do modelo AGA, denominada AGANd (Animação Gráfica baseada em Autômatos finitos Não determinísticos), sendo que o AGA utiliza apenas autômatos finitos determinísticos para criar os atores de uma animação. Com a utilização do AGANd obtém-se animações mais realistas e mais flexíveis.

A aplicação destas operações se dá nos dois modelos, os resultados obtidos a partir de cada uma das operações sobre os mesmos são apresentados de forma detalhada e ilustrados com os autômatos resultantes no decorrer do trabalho.

É apresentada uma sugestão de implementação para cada uma das operações, visando estender o protótipo já implementado do modelo AGA.

Isso faz com que o leitor seja estimulado a aplicar estas e outras operações categoriais em novas animações baseadas ou não nos modelos apresentados, despertando até mesmo para seu uso em outras áreas da Ciência da Computação.

Palavras-Chave: Teoria dos autômatos, Animação, Teoria das Categorias.

Investigation and Application of Categorical Operations among Actors of Animations based on Finite Automata with Output

ABSTRACT

This work presents an investigation on some graph-based categorical operations and the application of these operations to a specific field of Computer Science, that is, computer animation based on finite automata with output.

The categorical operations studied in this work are Product, Co-product, Pushout and Pullback. The AGA model (graphical animation based on finite automata) was chosen to be used as the basis of this dissertation.

Inspired in these studies, this work contains a proposal of how to apply such operations to define seemingly complex animations, in a simple, precise and easy to implement way. The focus is based on J. Stoy that says that one of the objectives for the use of the formal semantics of category theory is “to suggest means or forms for the planner to develop better, nicer (“clear”) systems, which have simpler formal descriptions”. However, it is not the objective of this work to verify if the use of these operations is or is not better than the use of any other solution to create new animations.

This dissertation brings a new version of AGA, called AGANd (Graphical Animations based on non-deterministic finite automata), instead of just using deterministic finite automata to create the actors of an animation as in the previous AGA model. With the use of AGANd we can get more realistic and more flexible animations.

The application of these operations is presented in the two models, the obtained results of each operation on them are presented in a detailed way and illustrated with the resulting automata.

An implementation suggestion for each one of the operations is also presented, which aims at extending the developed AGA model prototype.

That stimulate the reader to apply these and other categorical operations in new animations based or not on the presented models and also on other Computer Science areas.

Keywords: Automata Theory, Animation, Category Theory.

1 INTRODUÇÃO

Esta dissertação faz parte do contexto de um projeto de pesquisa denominado “Hyper Seed, Ferramentas e Métodos para Sistemas HiperMídia voltados para EAD via WWW”. Este projeto possui alguns desafios como:

- a) a unificação de modelos formais desenvolvidos em projetos de pesquisa do grupo (Hyper-Automaton, AGA, Nautilus);
- b) o desenvolvimento de um protótipo de sistema com suporte ao desenvolvimento de conteúdo multimídia e hiperMídia com ênfase em aplicações de aprendizado assistido por computador através da Internet (*e-learning*);
- c) a validação deste protótipo num contexto típico de instituição de ensino, junto ao Instituto de Informática da UFRGS;

Nos projetos Hyper-Automaton (e seus derivados *Hyper-Automaton: Interactive Evaluations*, XHA) e AGA, pode-se observar uma aproximação de abordagem no que se refere à utilização de autômatos finitos com saída (AFS) como modelo estrutural [MAH 2000, MAD 2002, MOR 2002, ACC 2002], conforme mostra a figura 1.1. Já o projeto Nautilus¹, apresenta sua abordagem focalizada na Teoria das Categorias [MEN2002a, STA2002] e na Orientação a Objetos (OO) [SEB 2001].

O projeto Hyper Seed pode ser lido na íntegra no Anexo A desta dissertação.

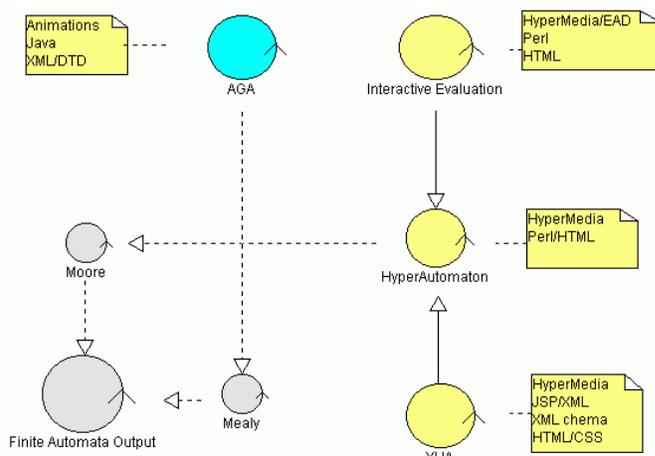


Figura 1.1: Pesquisas baseadas em modelos de autômatos finitos com saída

¹ Linguagem acadêmica de especificação orientada a objetos, concorrente e antecipatória, inspirada em Teoria das Categorias [MEN 2002b, MEN 2002, CAR 1999].

1.1 Motivação

O assunto “Teoria das Categorias” foi criado por S. Eilenberg e S. Mac Lane (1945 [MEN 2001]), em decorrência de seus trabalhos em topologia algébrica. Desde então, tem influenciado muitas áreas de conhecimento, como uma forma revolucionária de entendimento e de abordagem.

A Teoria das Categorias, na verdade, é um formalismo que pode ser visto como uma estrutura matemática em si, ou como formalização de estruturas matemáticas. À primeira vista pode parecer estranho aplicar Teoria das Categorias em Ciência da Computação, posto que 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: a independência de implementação (permite tratar propriedades abstratas “independentemente de implementação”), a dualidade (a noção categorial de dualidade divide o trabalho pela metade, tanto em termos de construções como de resultados), a herança de resultados e comparação de expressividade de formalismos (é possível herdar resultados já provados de uma categoria para outra, simplificando sobremaneira e até dispensando a verificação de alguns resultados), a notação gráfica (é a expressão de equações na forma de diagramas, facilitando a identificação e a compreensão de seus componentes e de seus relacionamentos) e a 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). [MEN 2001].

O propósito deste trabalho é estudar algumas operações categoriais e aplicá-las em Ciência da Computação, na área específica de animação gráfica baseada em autômatos finitos, verificando e relatando os resultados obtidos. É proposta uma nova forma de, a partir de dois ou mais atores do modelo original, aplicar uma operação categorial e construir, a partir daí, atores compostos. Os modelos originais não são perdidos, podendo assim retornar a utilizá-los a qualquer momento e recompô-los com uma nova construção categorial gerando um novo ator e uma nova animação.

1.2 Contextualização

Segundo J. Stoy [STO 77], um dos objetivos para o uso da semântica formal de teoria da categorias é “sugerir meios ou formas para o projetista desenvolver sistemas melhores, mais elegantes (“limpos”) e com descrições formais mais simples”, esse é o enfoque dado neste trabalho.

Vários estudos baseados em autômatos finitos já vem sendo efetuados. O trabalho de Machado [MAH 2000] foi o pioneiro na utilização dos autômatos para Hyper-Documentos na Web. Saito [SAI 2001] estudou uma nova forma de visualizar os autômatos de uma maneira mais clara, pretendendo com isto automatizar a tarefa de criação do autômato. Accorsi [ACC 2000, ACC 2001, ACC 2002] apresentou o modelo AGA - “Animações Gráficas baseadas na Teoria dos Autômatos”, criando animações computacionais baseadas em autômatos finitos, já com alguns protótipos implementados. Magalhães [MAH 2002], implementou um sistema de animação capaz de executar animações definidas de acordo com o modelo AGA, denominado AGA Player.

Nesse trabalho, é dada uma maior ênfase ao modelo AGA proposto por Accorsi . As aplicações categoriais estudadas são aplicadas sobre o modelo AGA e um novo modelo denominado AGANd (Animações Gráficas baseadas em Autômatos finitos Não determinísticos) que será descrito no decorrer desta dissertação.

O modelo AGA foi descrito por Accorsi e Menezes [ACC 2002] como um modelo para representar animações baseadas em Teoria dos Autômatos. O modelo AGA estrutura o conteúdo da animação em autômatos que descrevem o comportamento de atores durante o processo da animação. As características do AGA viabilizam a utilização de linguagens de consulta para recuperação de informações, ampliam as possibilidades de reutilização de partes da animação, assim como contribuem para a redução de espaço de armazenamento.

As vantagens do AGA, principalmente a redução do espaço de armazenamento, foram estudadas por Campani [CAM 2001] que, baseado na complexidade de Kolmogorov, fez uma comparação entre os modelos de animação AGA e GIF, conseguindo assim verificar as vantagens do primeiro sobre o segundo modelo, baseando-se nas principais características do modelo AGA citadas acima.

Além das vantagens estudadas e comprovadas por Campani, o que motivou a utilização do AGA como base para a aplicação de algumas construções categoriais foi o fato de que toda a sua construção é baseada em autômatos finitos. Vale a pena ressaltar que um autômato finito é um grafo finito, rotulado e com setas direcionadas [MEN 2000a, HOP 1979], portanto, todas as operações categoriais estudadas e aplicadas nas animações computacionais são baseadas em grafos.

O uso da Teoria das Categorias tende a dar uma visão “geral” do problema a ser tratado, não se preocupando com os detalhes (nos quais inclui-se a implementação), pois ela trabalha com operações poderosas o suficiente para tratar o problema a ser resolvido em alto nível. A aplicação da Teoria das Categorias entre atores de uma animação visa ressaltar características que são de fundamental importância em Ciência da Computação, como a independência de implementação, a dualidade, a herança de resultados e a expressividade de suas construções categoriais.

As operações categoriais de produto, coproduto, produto fibrado e soma amalgamada são interpretadas como operações composicionais sobre atores de uma animação, ou seja, construtores capazes de criar atores compostos a partir do modelo original.

A figura 1.2 traz o exemplo apresentado por Menezes e Machado [MEN 2000b], no qual um curso, baseado em autômatos finitos, é construído de forma incremental, utilizando-se os operadores categoriais. As caixas de traçado simples indicam os autômatos do curso original e as de traçado duplo os novos autômatos resultantes das operações.

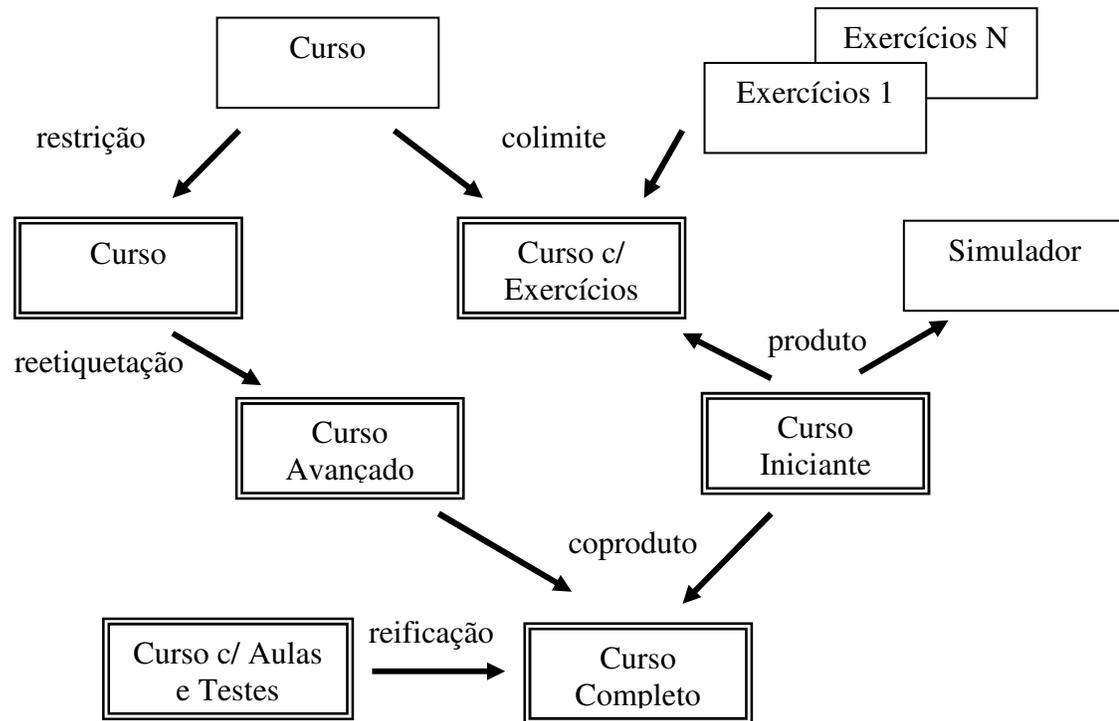


Figura 1.2: Construção de um curso

Conforme Menezes e Machado [MEN 99], mediante as categorias de autômatos é possível utilizar as ferramentas da Teoria das Categorias de forma a definir precisamente um conjunto de operações sobre os componentes de uma hiperbase.

O que se pretende neste trabalho é, tomando como base a construção do curso apresentado na figura 1.2, compreender a aplicação das construções categoriais e aplicá-las entre atores de uma animações verificando os tipos de atores que surgirão a partir dos atores originais.

1.3 Problemática

O problema encontrado é que o modelo AGA é baseado somente em autômatos finitos determinísticos e a forma de se fazer composições e combinações mais poderosas entre os atores muitas vezes requer modificações nas fitas de entrada ou nos autômatos originais.

1.4 Objetivo

Este trabalho tem como objetivo investigar e aplicar algumas operações categoriais entre atores de uma nova animação, o que poderá resultar na geração de novas animações baseadas em autômatos finitos, sempre utilizando o modelo AGA [ACC 2001] como base.

Afim de atingir esse objetivo foram traçadas as seguintes metas:

a) Estudar Teoria das Categorias com ênfase na categoria de grafos e definir as construções categoriais a serem interpretadas:

- Produto e Coproduto;
- Produto Fibrado e Soma Amalgamada;
- b) Criar uma extensão do modelo AGA, o AGANd (Animação Gráfica baseada em Autômatos finitos Não determinísticos), em que os atores da animação possam ser criados a partir de autômatos finitos não determinísticos.
- c) Aplicar estas construções categoriais no modelo AGA [ACC 2002] e no modelo AGANd.
- d) Sugerir uma implementação para cada uma das operações categoriais estudadas, estendendo o protótipo funcional do AGA.

1.5 Descrição e Apresentação da Dissertação

Este trabalho enfoca especificamente algumas construções categoriais com ênfase em grafos. Estas construções foram estudadas e suas respectivas teorias foram aplicada ao modelo AGA e AGANd verificando-se o resultado prático das mesmas. Essa dissertação está dividida da seguinte forma:

Capítulo 1 Introdução: apresenta a motivação, a contextualização, a problemática e os objetivos da presente pesquisa.

Capítulo 2 Projeto *Hyper Seed*: apresenta os principais aspectos do Projeto *Hyper Seed* e um resumo dos trabalhos já desenvolvidos no projeto como o Hyper Automaton, o Hyper-Automaton: Avaliação Interativa, o eXtensible Hiper-Automaton (XHA), a Animação Bidimensional para World Wide Web (AGA), o AGA Player e o Astrha.

Capítulo 3 Animações por computador: neste capítulo é apresentada uma pesquisa sobre animação por computador e as características desejáveis em animação para a Web. São apresentados também um resumo sobre animação bidimensional: Macromedia Flash (SWF), Scalable Vector Graphics (SVG), Graphics Interchange Format (GIF), Moving Picture Expert Group (MPEG-1) e Quick Time e Áudio Video Interleave (AVI).

Capítulo 4 Modelo AGA: são apresentados os principais pontos sobre o modelo AGA.

Capítulo 5 AGANd (Animação Gráfica baseada em Autômatos finitos Não determinísticos): apresenta uma extensão do Modelo AGA, com a utilização de autômatos finitos não determinísticos. Apresenta também uma solução para a implementação do não-determinismo, denominado AFNDp, que utiliza a noção de pesos nas arestas dos autômatos, sendo assim, devido aos pesos um dos caminhos do autômato é escolhido quando há o não determinismo.

Capítulo 6 Aplicação de Produto e Coproduto entre Atores de uma Animação: é apresentada a definição de Produto e Coproduto baseado em grafos, bem como os resultados obtidos com a aplicação dessas operações entre os atores de uma animação. É apresentado também o Produto e o Coproduto de Grafos Reflexivos e a aplicação de operações categoriais sobre operações categoriais. As fitas de entrada para cada uma destas operações categoriais é definida.

Capítulo 7 Aplicação de Produto Fibrado e Soma Amalgamada entre Atores de uma Animação: neste capítulo é apresentada a definição de Produto Fibrado e Soma Amalgamada em Grafos. O resultado obtido a partir da aplicação do Produto Fibrado como restrição de sistema, e como sincronizador de sistema, é apresentado quando aplicados sobre atores de uma animação. A Soma Amalgamada é vista sem a identificação das saídas e com a identificação das saídas e o resultado da aplicação da mesma sobre os atores de

uma animação é apresentado. A fita de entrada para cada uma destas operações categoriais é definida e um exemplo é apresentado.

Capítulo 8 Sugestão de Implementação: neste capítulo é dada uma sugestão de implementação para cada uma das operações categoriais estudadas.

Capítulo 9 Animação composta por Operações sobre Operações: neste capítulo é apresentado um exemplo de uma animação, a princípio considerada complexa, e sua resolução utilizando-se as operações categoriais que foram investigadas.

Capítulo 10 Conclusão e Trabalhos Futuros: reflete sobre os resultados obtidos no estudo feito para esta dissertação e propõe trabalhos futuros.

Anexo A: Projeto de pesquisa “*Hyper Seed – Framework*, Ferramentas e Métodos para Sistemas Hipermídia voltados para EAD via WWW”, proposto pelo Instituto de Informática da UFRGS em 12/11/2002, aprovado e apoiado pelo CNPq, do qual esta dissertação faz parte.

Anexo B: Artigo “*Autômato Finito Não-determinístico com pesos aplicado a animações baseadas em autômatos finitos*”. Revista Científica da UNESC (Universidade do Extremo Sul Catarinense, em julho de 2002.

Anexo C: Projeto de pesquisa e desenvolvimento Tecnológico em Software Livre “*PNEE – Autômato: Sistema Hipermídia para Portadores de Necessidades de Educação Especial, baseado em Teoria dos Autômatos*”, proposto pelo Instituto de Informática da UFRGS em 29/10/2003, para aprovação pelo CNPq.

Referências

2 PROJETO HYPER SEED

2.1 Introdução

Este capítulo traz um resumo dos estudos que estão sendo feitos dentro do Projeto *Hyper Seed*, Ferramentas e Métodos para Sistemas Hiperfídia voltados para EAD via WWW. De uma forma geral, essas pesquisas têm dado preferência à estruturas de máquinas de Mealy² em relação às máquinas de Moore³ visto à tendência das máquinas de Mealy resultarem em máquinas com menor quantidades de estados do que equivalentes máquinas de Moore, devido à semântica associada às transições ou aos estados, respectivamente.

O anexo I apresenta, na íntegra, o projeto *Hyper Seed*. Para que se tenha um melhor esclarecimento do que está sendo feito, sugere-se uma visita à página do grupo em <http://teia.inf.ufrgs.br> [MEN 2003].

2.2 Hyper-Automaton

Em seu trabalho de dissertação, Machado [MAH 2000] procurou validar o uso de AFS para estruturar sistemas hipertextos, em especial para cursos na Web, apresentando o *Hyper-Automaton*: Hipertextos e Cursos na Web Usando Autômatos Finitos com Saída. Machado diz que sua pesquisa está centrada nos seguintes aspectos:

[...]estudo da aplicação do formalismo de Autômatos Finitos com Saída (Máquina de Mealy e Máquina de Moore) como modelo estrutural para a organização de hiperdocumentos instrucionais, em especial de Cursos na Web. O modelo é inspirado por pesquisas clássicas na área de hiperdocumentos e recentes iniciativas na WWW, com especial enfoque no desenvolvimento de sistemas hipertexto onde a base de hiperdocumentos é projetada de forma independente da estrutura de controle da aplicação hiperfídia, e suporta algumas facilidades descritas no Modelo Dexter como a composição de estruturas hierárquicas, especificação de vários conjuntos de links sobre um mesmo corpo de hiperdocumentos e objetos separados da estrutura de navegação [MAH 2000].

O funcionamento do Hyper-Automaton, conforme a página do grupo de pesquisa do *Hyper Seed* [MEN 2003], esta definido da seguinte forma:

² Autômato finito modificado de forma a gerar uma palavra de saída a cada transição [MEN 2000].

³ Autômato finito com saída que gera uma palavra de saída (que pode ser vazia) para cada estado da máquina [MEN 2000].

Cada autômato define um curso e consiste em um conjunto de hiperdocumentos independentes, os quais podem pertencer a outros cursos. A função de transição funciona como ligação lógica entre os hiperdocumentos e a função de saída compõe as páginas. O resultado final é a estrutura básica de páginas e links de hipertexto em um site na Web.

A interface do ambiente de um navegador Web fornece uma interpretação tangível para a estrutura de autômatos em hiperdocumentos. Os símbolos do alfabeto de saída são anotados com unidades de informação (páginas HTML) e, neste caso, o resultado das funções de saída (Máquina de Moore) ou de transição (Máquina de Mealy) é a apresentação de um hiperdocumento na janela do navegador, de acordo com os fragmentos indicados na palavra de saída.

O alfabeto de entrada, que nomeia as transições do autômato, são apresentados ao usuário como âncoras de links de navegação.

O sistema Hyper-Automaton também fornece funcionalidade para a construção de avaliações adaptativas via Internet. Neste caso, entende-se como adaptativa, uma prova cuja seqüência de apresentação de questões é dependente da performance do aluno ao responder questões prévias.

Nos exercícios e provas disponibilizados no sistema, as questões foram construídas segundo três níveis de dificuldades e o aluno responde primeiramente uma questão de nível médio, caso acerte a resposta o sistema providencia uma questão de nível mais alto, caso contrário terá uma chance de recuperar o conceito errôneo através de uma questão formulada especialmente para a recuperação.

O modelo leva a um alto grau de modularização do material instrucional, apresentando as seguintes vantagens:

1. Facilidade de reuso de páginas em diversos cursos, com eliminação da redundância.
2. Independência dos hiperdocumentos da estrutura do autômato, cuja alteração não influi nas páginas e vice-versa.
3. Permite que qualquer usuário crie links de e para qualquer documento.
4. Facilidade de implementação e manutenção; interface gráfica simples e direta.
5. Elaboração de seqüências instrucionais com enfoques específicos e capaz de oferecer estudo individualizado.
6. Operações categoriais fornecem um esquema de composição de cursos que permite a construção de novos cursos sobre cursos já existentes através de procedimentos de alto nível .

2.3 Hyper-Automaton: Avaliação Interativa

O trabalho de Morais [MOR 2002] apresenta uma versão especializada do *Hyper-Automaton* destinada a automatizar a aplicação de avaliações de conhecimentos, o *Hyper-Automaton: Avaliação Interativa de Alunos em Cursos na Web Baseado em Autômatos Finitos*. Segundo Morais seu trabalho trata-se de:

[...]uma técnica da aplicação do formalismo de Autômatos Finitos com Saída (Máquina de Mealy e Máquina de Moore) como um modelo estrutural para a organização de hiperdocumentos instrucionais, em destacar especial, Avaliação e Exercício. Esse objetivo é motivado pela organização e agilização do processo de avaliação proporcionado ao professor e ao aluno.

Ainda segundo Morais, o funcionamento das provas adaptativas é descrito da seguinte forma:

Cada prova é composta de seis questões objetivas, de cinco opções, sendo que apenas uma está correta. Responde-se as questões na ordem em que desejar. Ao final, deve-se clicar em enviar; preenchendo-se o campo de nome e clicar em enviar novamente.

Todas as questões devem ser respondidas antes de enviar os dados, caso contrário, será pedido que as responda. O resultado da aplicação da prova é imediato, mostrado ao aluno pouco depois do envio.

Nestes testes eletrônicos é perfeitamente aceitável o uso de material de consulta, seja através de livros, seja pela internet.

2.4 eXtensible Hyper-Automaton (XHA)

Outro trabalho da mesma linha de pesquisa foi apresentado por César Machado e Gustavo Federizzi [MAD 2002], denominado de: *eXtensible Hyper-Automaton (XHA)*. Segundo os autores esse trabalho tem como objetivo:

[...]estudar as possibilidades de flexibilização da função de saída do Sistema Hyper-Automaton além das rígidas possibilidades utilizadas atualmente com a utilização direta do HTML, objetivando eliminar as limitações como execução de aplicações proprietárias, caracteres incompatíveis entre browsers, excesso de tráfego na rede, padronizar aplicações, incrementar recursos didáticos, melhorar o suporte a aplicações multimídia atuais e futuras, facilitar a manutenção, implementação e reuso, alterar o layout de saída no browser de maneira dinâmica, explorar outros recursos de links, estabelecer padrões de organização do material instrucional criado pelo professor e muitas outras. Tal sistema anteriormente desenvolvido e funcionando adequadamente, é baseado no formalismo de Autômatos Finitos com Saída como modelo estrutural para organização de hiperdocumentos instrucionais, em especial em cursos na Web, tornando o material hipermídia independente do controle da aplicação. O Sistema Hyper-Automaton tornou-se, portanto, um sistema semi-automatizado para suporte a cursos na Web.

Nas conclusões dos autores, foi realizada uma comparação do XHA com a implementação anterior do Hyper-Automaton, e foram verificados os seguintes itens:

- **Organização de arquivos:** Na versão anterior cada curso tinha seu próprio diretório onde ficavam os seus três arquivos de definição: curso.fl, alfabeto.fl e estados.fl. Além disso, todos os fragmentos HTML disponíveis eram armazenados em apenas um diretório do servidor;
- **Manutenção facilitada:** Enquanto que, na versão codificada em Perl, o acesso às máquinas de Moore e de Mealy eram feitos através de dois sistemas diferentes, na nova versão em JSP a funcionalidade das duas máquinas foram incorporadas em um só sistema. Assim, a manutenção se torna muito mais fácil e o acesso ao sistema se torna único;
- **Reuso inteligente:** O primeiro sistema Hyper-Automaton já utilizava o conceito de reuso através da disponibilização de material instrucional na forma de fragmentos de HTML. Através do uso de XML e XSL, o conteúdo do material de ensino foi separado de sua forma de apresentação. Isso significa que um mesmo objeto de ensino é apresentado conforme a diagramação do curso que o utiliza.

2.5 Animação Bidimensional para World Wide Web (AGA)

Do mesmo grupo de estudos, surgiu outra pesquisa que utilizou AFS como elemento estrutural em outra área da ciência da computação, a da Computação Gráfica. Fernando Accorsi visualizou a aplicação dos AFS como *framework* para a definição de animações por computador e apresentou o AGA: Animação Bidimensional para *World Wide Web* Baseada em Autômatos Finitos [ACC 2002].

Esse trabalho será mais detalhado posteriormente por se tratar do trabalho que será utilizado como base para o estudo dessa dissertação.

Accorsi definiu os objetivos do AGA no seguinte parágrafo:

Este trabalho aplica a Teoria de Autômatos na proposição de uma nova alternativa para prover animações 2D na World Wide Web, verificando as contribuições alcançadas para as questões relacionadas ao espaço de armazenamento, reutilização e manutenção do conteúdo e suporte à recuperação de informação. Para este objetivo, é proposto o modelo AGA (Animação Gráfica baseada em Autômatos Finitos), o qual especifica a animação a partir de uma estrutura baseada em autômatos finitos com saída. Esse modelo é definido de tal forma que os mesmos autômatos utilizados na especificação, ao serem simulados, realizam o controle da animação durante a apresentação. O modelo AGA apresenta características que favorecem a redução do espaço de armazenamento da animação, provêm suporte à recuperação de informação, colaboram com a reutilização e manutenção do conteúdo das animações.

Entre os vários conceitos e características do projeto de pesquisa AGA, pode-se destacar:

- **Atores:** Uma animação em AGA é uma lista de atores sincronizados, modelados através de máquinas de Mealy, cada ator contendo sua própria fita de entrada, que irá determinar seu comportamento.
- **Máquina de Mealy:** Como dito no item acima, o comportamento de cada ator é definido através de uma Máquina de Mealy determinística onde 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.
- **Controle de Tempo na Fita de Entrada:** A fita de entrada de cada autômato (Máquina de Mealy) deve definir quanto tempo cada imagem de ator deve ser apresentada, sendo assim um parâmetro de sincronização do ator com a animação.
- **Transformação de imagens:** A fita de entrada também pode definir transformações de imagens do tipo rotação, translação, zoom e visibilidade.
- **Reutilização:** Através do AGA, é possível mudar as ações de uma ator apenas mudando sua fita de entrada. Essa característica mais as transformações de imagens permitem a geração de animações longas e variadas com um conjunto pequeno de objetos gráficos e sonoros. As características de reutilização de AGA permitiram, em estudos de caso realizados em laboratório, aprimoramentos substanciais em relação a outras tecnologias.

2.6 AGA Player

Outro trabalho também na mesma linha de pesquisa é o AGA Player, desenvolvido por Magalhães [MAH 2002], como projeto de diplomação. Segundo o autor:

O programa de visualização, AGA Player, foi desenvolvido em linguagem Java na forma de applet. A escolha pelo Java foi motivada principalmente pela necessidade de portabilidade do visualizador e o amplo repositório de objetos existentes para manipulação gráfica e processamento do XML encontrado na linguagem. Para prover os recursos gráficos necessários para o processamento das imagens e sons foi utilizada a plataforma Java 2 [JAV 02], em especial, os componentes de interface gráfica com o usuário (GUI) do Swing⁴. A manipulação da linguagem AgaML no applet é realizada a partir de uma API destinada ao processamento de documentos XML específica para Java, o JDOM [JDO 02]. O JDOM pode ser visto como uma alternativa para o DOM (Document Object Model) e o SAX (Simple API for XML) direcionada para programadores Java.

O AGA Player é executado no cliente para realizar a reprodução da animação com base nas especificações em AgaML. A execução do AGA Player pode ser dividida esquematicamente em três fases:

- a) **Carga do AgaML e mídias externas** – O *applet*, ao ser inicializado, carrega o arquivo em AgaML cujo nome foi passado como parâmetro. Uma vez carregado, o arquivo AgaML é processado, e as informações das especificações dos atores e fitas são extraídas. Entre as informações estão as referências para os arquivos de imagens e sons utilizados, os quais são carregados na seqüência, e armazenados em estruturas de dados do *applet*. Como as mídias se encontram em arquivos independentes, o processo de carga aguarda a confirmação de leitura de todas antes de iniciar as outras fases;
- b) **Construção das instâncias** – A partir das especificações das instâncias extraídas do AgaML, os atores AGA são efetivamente instanciados em estruturas de dados internas e associados às suas respectivas fitas de entrada. Após esta fase, as instâncias se encontram no estado inicial e prontas para serem simuladas;
- c) **Simulação dos autômatos e geração dos quadros** – Os autômatos são simulados em paralelo para obter a formação dos quadros de saída. Porém, a implementação do modelo utilizando processos concorrentes para cada autômato (*multithreading*) implica em uma dificuldade adicional para a sincronização precisa das trocas de imagens dos autômatos. Assim, uma alternativa de implementação foi escolhida para garantir que as imagens geradas representem fielmente as especificações determinadas pelas fitas de entrada. Um processo de temporização (*timer*) foi criado para gerar eventos de análise dos autômatos com a frequência determinada pelo *FRAMERATE*. A cada evento do *timer*, as transições pertinentes são realizadas em cada um dos autômatos e o quadro de saída é formado, reproduzindo assim a imagem respectiva para aquele instante na animação. Durante o processo de simulação, o AGA Player está disponível para intervenções de reprodução e consulta solicitadas pelo observador através dos botões de controle.

⁴ Pacote que contém componentes escritos, manipulados e exibidos completamente em JAVA, denominados comumente de componentes Java puros.

A figura 2.1 ilustra o painel de controle do AGA Player visualizado pelo observador durante visita à página *Web*. O painel conta com 3 botões de controle de reprodução para as funções corriqueiras de reproduzir, pausar e parar. Uma barra de busca é disponibilizada para o usuário com o objetivo de indicar o tempo decorrido de animação e proporcionar a busca temporal da seqüência animada.

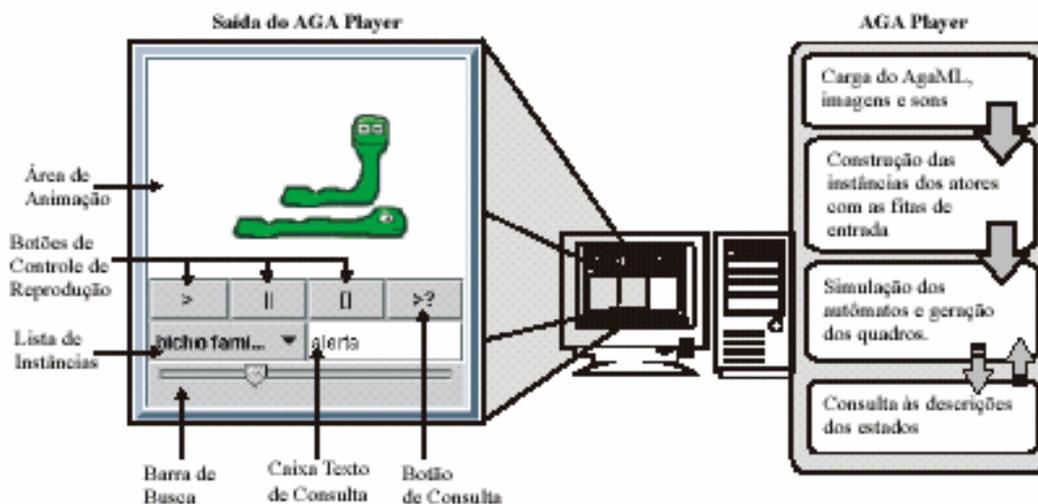


Figura 2.1: Painel de controle do AGA Player visualizado pelo observador

O AGA Player conta com um recurso de busca incomum nos programas de visualização disponíveis na Internet. Este recurso é particular do modelo AGA e proporciona um mecanismo refinado de pesquisa baseado em palavras-chave. A lista de instâncias, a caixa de texto e o botão de consulta dão suporte operacional para a execução deste mecanismo.

O procedimento de pesquisa é operacionalmente simples. O observador escolhe um ator na lista de instâncias e digita alguma palavra-chave que represente a situação semântica que deseja verificar. Assim, quando o botão de consulta é pressionado, é realizada uma análise na instância, investigando se em algum instante da animação ela ocupa um estado cuja descrição vinculada coincide com a palavra-chave requerida. Caso afirmativo, a animação é posicionada neste instante, pronta para iniciar a reprodução. A análise é realizada a partir do instante corrente. Desta maneira, o botão pode ser pressionado sucessivas vezes a fim de pesquisar novas situações ao longo da animação. A potencialidade de casamento das palavras-chave cresce à medida que o projetista da animação associa um amplo vocabulário aos estados através da função descrição.

O mecanismo de consulta implementado no AGA Player demonstra como o modelo AGA pode favorecer a recuperação de informação em animações. Esta implementação, porém, não esgota a potencialidade do modelo para este propósito. É possível implementar mecanismos de consulta baseados na construção de predicados mais complexos que levem em conta várias instâncias dos atores AGA. A estrutura do AgaML promove a reutilização das especificações em vários níveis. As especificações dos atores e fitas podem ser utilizadas por várias instâncias diferentes, assim como, podem ser compartilhadas por diversas animações se armazenadas em arquivos independentes. Essa última característica colabora também para que as especificações possam ser criadas dinamicamente por processos vinculados às páginas no momento do acesso pelo usuário.

2.7 Astrha

Astrha é uma ambiente Gráfico, Dinâmico e Interativo para Internet Baseado em Hiper-Animações e na Teoria dos Autômatos, que faz parte do projeto Hyper Seed, e foi desenvolvido por Grandi [GRA 2003]. Segundo o autor:

A unificação dos conceitos de sistema hipermídia e animações através do conceito de hiper-animação permitiu que as características mais relevantes das especificações Hyper-Automaton, Hyper-Automaton: Avaliações Interativas, eXtensible Hyper-Automaton (XHA) e Animação Bidimensional para World Wide Web (AGA) fossem unificadas em nível conceitual e estruturadas através de uma definição sintática de Máquina de Mealy (Astrha/M), especializada para esse fim. A linguagem Astrha/L permite que os desenvolvedores de códigos nessa linguagem especifiquem, de forma simples e direta, hiper-animações, uma vez que a linguagem é de quarta geração. Tanto Astrha/M como Astrha/L definem máquinas de Mealy reflexivas, com conjunto finito de palavras de saída, não-determinísticas em nível de estado e de palavra de saída. No protótipo Astrha/E a reflexividade possui semântica de inoperabilidade aparente (NOP) e o não-determinismo de escolha pseudo-aleatória, dando poder de programação ao desenvolvedor Astrha/L através da definição de transições convenientes, nas máquinas de Mealy, para a solução de problemas.

Esse trabalho possui, entre seus objetivos e metas:

- a) o desenvolvimento de uma fundamentação matemática para a unificação, de maneira coerente e matematicamente rigorosa, de especificações de sistemas hipermídia e animações baseadas na Teoria dos Autômatos;
- b) a construção e validação de um protótipo de sistema com suporte à criação de conteúdo multimídia e hipermídia com ênfase em educação assistida por computador;
- c) a definição e aplicação de estudos de caso.
- d) atender às demandas acadêmicas e construtoras supra citadas, no que se refere à unificação de especificações de sistemas hipermídia e animações baseadas na Teoria dos Autômatos, em nível conceitual, é o objetivo principal do Astrha. Mais especificamente, unificar conceitos das especificações *Hyper-Automaton*; *Hyper-Automaton: Avaliações Interativas*; *eXtensible Hyper-Automaton (XHA)* e *Animação Bidimensional para World Wide Web (AGA)*.

2.8 Considerações Finais

O Projeto *Hyper Seed* agrega uma grande quantidade de pesquisas e trabalhos, todos em uma mesma linha de raciocínio que é a de desenvolver uma fundamentação matemática para a unificação, de maneira coerente e (matematicamente) rigorosa, de especificações de textos e animações, baseada na Teoria dos Autômatos e das Categorias. É nesta linha que esta dissertação também está incluída. O que se pretende com esta pesquisa é utilizar Teoria das Categorias como fundamento matemático para que se possa fazer composições entre os atores originais de uma animação a fim de se conseguir uma nova animação. Tudo isso deve ocorrer sem que se perca a animação original e utilizando-se a noção de reuso das partes de um ator ou até mesmo do todo.

Tudo estará fundamentado matematicamente com a utilização das operações categoriais. Este trabalho servirá como uma fonte de pesquisa para os demais trabalhos que serão desenvolvidos no Projeto *Hyper Seed*.

3 ANIMAÇÕES POR COMPUTADOR

3.1 Introdução

Segundo Thalmann, uma animação por computador consiste em “modificar uma cena no tempo [THA 91].” Cenas tridimensionais, por exemplo, podem ser compostas por três tipos de entidade: objetos, câmaras e luzes. Uma animação também pode ser definida como um processo no qual cada quadro de vídeo⁵ em um filme é produzido individualmente, causando-nos uma sensação de movimento. Para que nossos olhos e cérebros percebam animações como movimentos contínuos e ininterruptos necessita-se que as seqüências de quadros seja atualizada a uma taxa de, pelo menos, 30 quadros por segundo. Abaixo dessa taxa, a maioria das pessoas podem detectar *flickers*⁶ que depreciam o realismo. Em um limite superior, taxas acima de 70 quadros por segundo, o ser humano não percebe melhorias devido às características funcionais dos olhos e do cérebro. A razão pela qual acima desta taxa *flickers* não são percebidos deve-se à persistência da visão (*persistence of vision*). De tempo em tempo, o cérebro armazena as imagens recebidas através dos olhos por uma fração de segundo e, automaticamente, atenua sobressaltos menores [WIK 2002].

A animação convencional⁷ é definida em [THA 85] como a técnica na qual cria-se a ilusão de movimento através de uma série de desenhos fotografados individualmente e gravados em sucessivos quadros em um filme. A ilusão de movimento é alcançada quando o filme é projetado a uma certa taxa de quadros por segundo. Animação também se refere ao processo de geração dinâmica de uma série de quadros representando um conjunto de objetos, no qual cada quadro é uma alteração do quadro anterior.

A animação por computador teve sua origem na assistência dos processos convencionais e atingiu sua independência nas animações modeladas. A animação assistida por computador consiste, basicamente, em automatizar processos empregados na criação de animações convencionais, enquanto a animação modelada engloba desde a modelagem geométrica dos objetos até a computação das diferentes ações que podem realizar na animação.

Existem várias técnicas para se obter uma animação, com ou sem o uso de computadores. Nos casos de uso de computadores, são objeto de estudo especializado especialmente na área da computação gráfica. O processo de construção de uma animação computadorizada pode incluir as seguintes etapas [PUE 88, THA 91, THA 85, FOL 90]:

⁵ Um quadro (*frame*), no contexto das animações, corresponde a uma imagem completa.

⁶ Em processamento de imagens, efeitos de tremulação, vibração ou perda de características que ocorrem em um dispositivo de saída visual devido à baixa taxa de renovação da imagem ou corrupção de sinal.

⁷ O termo “convencional” é utilizado para as animações tradicionalmente criadas em sistemas não computacionais.

- a) **Pré-processamento** – A animação é planejada através de *storyboards* onde as seqüências animadas são esboçadas a partir das cenas principais. Essa fase é amplamente usada em filmes de animação devido à complexidade do projeto, já em animações mais simples geralmente é descartada.
- b) **Edição das cenas e modelagem dos objetos** – Os objetos que participam da animação são modelados e combinados para comporem as cenas. A modelagem dos objetos pode ser realizada através de várias abordagens dependendo da natureza estética. Em geral, as modelagens mais comuns são as geométricas em domínios bidimensionais e tridimensionais através de modelos baseados em arames ou sólidos [THA 85].
- c) **Animação** – As variações dos objetos na animação são determinadas e calculadas para gerar a seqüência animada. Os movimentos dos objetos na animação podem ser descritos através de diversos métodos de controle movimento e em níveis de abstração diferentes. Esses métodos vão desde o controle explícito, no qual o animador é responsável pela descrição dos atributos posicionais dos objetos, até controles altamente automatizados através de sistemas baseados em conhecimento [FOL 90].
- d) **Rendering das imagens** – As imagens finais de cada quadro são obtidas através da computação das propriedades visuais dos objetos na cena. Nesta fase são levados em conta, por exemplo, os parâmetros de iluminação e posicionamento de câmeras para a computação de sombras, transparências, faces ocultas, texturas, etc. [THA 85] salienta que o processo de *rendering* pode ser realizado em tempo real durante a apresentação da animação ou ser computado antes.
- e) **Pós-processamento** – A seqüência de imagens geradas é sincronizada com som e gravada em outro tipo de mídia caso necessário. Nas animações destinadas a filmes, a seqüência é combinada com outras cenas e gravada em película.
- f) **Análise dos resultados** – As características visuais reveladas pela animação são exploradas com o objetivo de extrair informações sobre os modelos utilizados. Essa fase é amplamente utilizada em animações destinadas à simulação de processos naturais e industriais.

A montagem de uma animação por computador possui etapas semelhantes a um projeto convencional de software. O pré-processamento pode ser comparado à análise preliminar. A edição de cenas e a modelagem de objetos nas animações equivalem à etapa de projeto do software. A animação e a renderização correspondem à fase de construção do software. O pós-processamento tem semelhanças com o processo de integração de módulos e a análise de resultados com a busca de informações elaboradas.

Os sistemas responsáveis por essas tarefas, chamados sistemas de animação, possuem três componentes básicos: modelador de atores⁸, mecanismo de controle da animação e mecanismo de *rendering* e visualização.

Devido à variedade de técnicas utilizadas no processo de animação, vários enfoques para a classificação dos sistemas de animação são propostos [ZEL 85, THA 85, FOL 90, PUE 88]. Em [PUE 88] são reunidos os seguintes critérios para classificação: histórico, tipo de aplicação, nível de controle do movimento no sistema, dimensão do sistema, modelo de animação, complexidade de *rendering*, complexidade de pós-processamento e modo de produção.

⁸ Os objetos que participam da animação são comumente chamados de atores.

O critério baseado na dimensão do sistema, apresentado em [PUE 88], estabelece a classificação dos sistemas de animação em 2, 2.5 e 3 dimensões. A dimensão do sistema está diretamente ligada à natureza geométrica da modelagem dos objetos e da construção da cena. Em sistemas bidimensionais, as modelagens dos objetos e das cenas são realizadas sobre um espaço bidimensional. Em animações 2.5 dimensionais, os objetos bidimensionais são distribuídos em um ambiente tridimensional, onde transformações geométricas tridimensionais são permitidas e a cena é construída a partir da computação de um campo de visão sobre este ambiente. Já em sistemas tridimensionais, tanto os objetos quanto à construção das cenas são modelados em um espaço tridimensional.

Independente da dimensão, os sistemas de animação por computador também podem ser classificados, segundo o critério de modo de produção, em sistemas de tempo real ou quadro a quadro [THA 85]. Os sistemas de animação em tempo real geram a imagem final para visualização no momento de sua apresentação. Esta abordagem favorece principalmente as animações interativas, onde a imagem visualizada deve corresponder às ações instantâneas tomadas pelo usuário. Por outro lado, os sistemas de animações quadro a quadro, geram a seqüência animada a partir de seus modelos de objetos e cenas, armazenando as imagens geradas em sucessivos quadros. Posteriormente, esses quadros são apresentados a uma taxa adequada para visualizar a animação. Em geral, esta abordagem é utilizada quando a complexidade de *rendering* é alta devido ao realismo das cenas.

3.2 Características da Web

Segundo Lew [LEW 2000], grande parte do conteúdo da Web consiste atualmente em informações visuais, tais como imagens, animações, vídeos e gráficos. As animações, em especial, são empregadas para diversos fins, como por exemplo, as propagandas *on-line*, jogos, ícones animados, ambientes virtuais, interfaces gráficas, desenhos animados, simulações, entre outros.

Os sistemas de animação para Web materializam a seqüência animada em um arquivo para o intercâmbio da animação na rede. Este arquivo é utilizado para a apresentação da animação pelos módulos de visualização localizados, em geral, nos navegadores. O arquivo de animação e o módulo de visualização são os componentes que interagem diretamente com o ambiente, as demais partes do sistema de animação são, na maioria das vezes, independentes.

Há diversos formatos de arquivos⁹ para o intercâmbio de animações na Web. Vários, já eram utilizados anteriormente em outros ambientes e foram adaptados às necessidades da WWW. Outros formatos, porém, foram desenvolvidos especialmente para este ambiente, considerando as características de rede, a heterogeneidade das plataformas, a dinâmica do conteúdo, entre outras.

Em [LEE 98], os formatos de arquivos para animação são classificados em formatos baseados em quadros ou conteúdo. A primeira categoria armazena as animações como uma seqüência de imagens estáticas referentes aos quadros. Já a segunda, armazena os modelos que descrevem os objetos e suas ações ao invés das imagens. O GIF, MPEG-1 [INT 96], QuickTime [APP 2000] e AVI [MIC 2002] são exemplos de formatos baseados em quadros herdados pela Web, enquanto o SVG [W3Ca 2002] e o SWF [MAM 2002] são propostas desenvolvidas especificamente para este ambiente baseadas no conteúdo.

⁹ Formato neste contexto refere-se à especificação para armazenamento da animação em arquivos de dados.

3.3 Animações bidimensionais

3.3.1 Macromedia Flash (SWF)

Flash é uma linguagem desenvolvida pela empresa *Macromedia* que utiliza vetores gráficos para definição de objetos, sendo o padrão de fato para a Internet utilizando-se esta abordagem computacional. Suporta interatividade, textos, imagens, movimentos na tela. Esse sistema tem sido utilizado comumente para a criação de interfaces gráficas diferenciadas para sítios, produção de seqüências animadas com baixo nível de realismo e no desenvolvimento de jogos *on-line*.

As animações em Flash podem conter atores gráficos baseados em curvas ou mapa de bits, textos, vídeos, e objetos de interação. O sistema permite a criação de seqüências animadas interativas, assim como a sincronização destas com efeitos sonoros.

O ambiente de criação conta com um modelador geométrico de atores com diversas ferramentas de desenho e colorização, um mecanismo de animação baseado na interpolação de quadros-chave, um editor para a linguagem de script, geradores para diversos formatos de arquivo de intercâmbio e um sistema integrado de visualização das animações. O modelador de atores provê ferramentas de modelagem baseadas em primitivas geométricas e desenho livre. Os desenhos modelados são convertidos em representações vetoriais e colorizados a partir de esquemas de cores sólidas e gradientes. Os textos são editados através de uma ferramenta especial e podem ser manipulados como figuras vetoriais. Os vídeos e imagens na forma de mapa de bits podem ser inseridos no ambiente a partir de mecanismos de importação de arquivos. Os objetos gráficos podem ser manipulados na área de trabalho a partir de transformações geométricas como escala, translação e rotação. O ambiente ordena a sobreposição dos objetos através da organização da área de trabalho em camadas.

O controle de animação utilizado pelo Flash é explícito. O animador define as alterações através de transformações geométricas dos atores e mudanças de atributos de colorização. O mecanismo de animação também provê processos de interpolação de quadros-chave, no qual o animador define interativamente os parâmetros dos atores nos quadros-chave origem e destino, deixando para o sistema a criação dos quadros intermediários. Os movimentos também podem ser definidos através de scripts escritos na linguagem específica do sistema, *ActionScript*. Essa última alternativa de controle de movimento é bastante utilizada na criação de animações interativas.

A seqüência animada pode ser codificada em diversos formatos de arquivos, como o GIF, AVI e QuickTime, porém, o SWF, padrão do sistema Flash, é o formato de arquivo que comporta a especificação de todos recursos previstos pelo ambiente. O SWF é lido como mídia contínua pelo *Macromedia Flash Player*, *plug-in* de visualização disponibilizado pela mesma empresa.

A interatividade com o usuário é realizada através de códigos escritos em *ActionScript*, linguagem *script*¹⁰ desenvolvida especialmente para comunicação com a linguagem *Flash*.

¹⁰ O termo script é aplicado a linguagens fracamente tipadas, geralmente interpretadas, que possuem estruturas de dados simples que servem, tipicamente, como meio de interação (*interface*) com outras linguagens [DCI 2003, MAC 2002a, p. 53].

Uma técnica de computação gráfica bastante utilizada por desenvolvedores *Flash* é a *tweening*¹¹.

Grande parte da popularidade do Flash está atribuída à eficiência do SWF como formato de intercâmbio de animações. O SWF foi desenvolvido especificamente para a distribuição de gráficos vetoriais e animação na Internet. Um arquivo SWF pode ser visto como um conjunto de blocos etiquetados distribuídos sequencialmente. Este conjunto é sempre precedido por um cabeçalho de identificação e concluído por um bloco especial de finalização. Os demais blocos são categorizados entre blocos de definição ou controle. Os blocos de definição especificam as figuras, textos, imagens e sons utilizados na animação, enquanto os blocos de controle determinam as variações dos objetos e o controle de fluxo da animação. Com objetivo de apresentar a animação como mídia contínua, o conteúdo de cada bloco só depende de blocos anteriores a ele. A figura 3.1 mostra um exemplo de sequência de blocos do SWF.

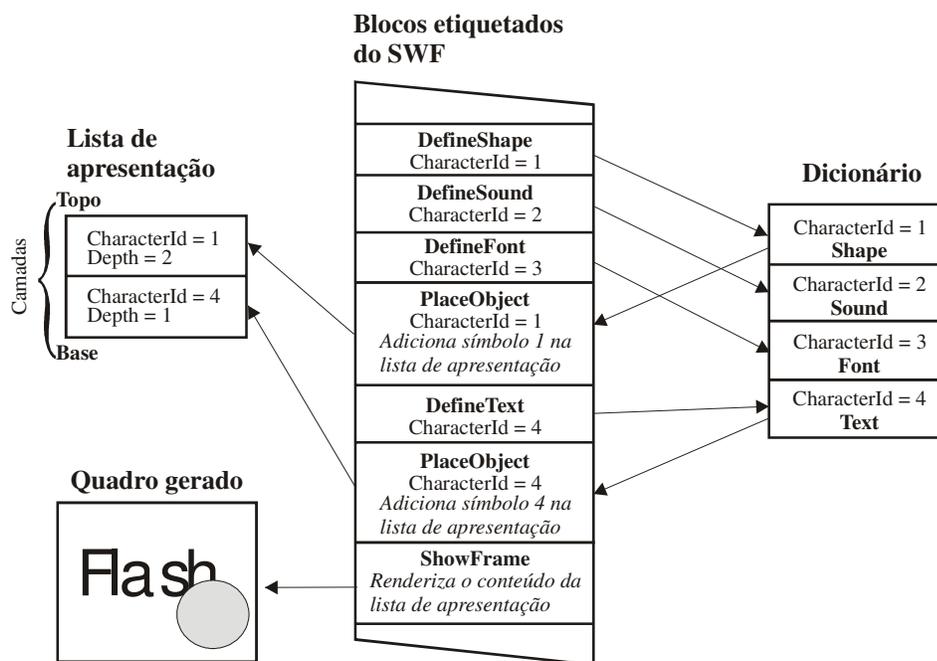


Figura 3.1: Processamento do SWF no programa de visualização

O SWF também é utilizado pelos sistemas de animação Corel R.A.V.E. e Adobe Live Motion como formato padrão para o intercâmbio de animações. Estes sistemas são bastante similares ao Flash.

3.3.2 Scalable Vector Graphics (SVG)

¹¹ Técnica de interpolação na qual um programa de animação gera automaticamente quadros intermediários (denominados *in between*) entre quadros chaves (*key frames*). Esta técnica foi introduzida por Burtnyk e Wein em 1971 [THA 89, DCI 2003].

O SVG é um vocabulário¹² proposto pelo W3C para representar gráficos bidimensionais, estáticos ou animados, em XML. O alto nível das estruturas sintáticas e semânticas da linguagem vem possibilitando não só sua utilização como linguagem alvo em sistemas de animação, mas também como linguagem fonte destinada à edição humana. O SVG, somado a tecnologias como o ECMAScript¹³ e SMIL [W3Cb 02], tem se mostrado um próspero concorrente do Flash (SWF). Um arquivo em SVG consiste em um documento no formato texto estruturado segundo as marcações definidas pela DTD. Esse arquivo texto é utilizado como arquivo de intercâmbio na Internet e processado pelos programas de visualização, em geral, *plug-ins* instalados nos navegadores. A DTD do SVG define tipos de elementos¹⁴ para manipulação de três categorias de objetos gráficos: figuras vetoriais, imagens mapa de bits e texto. A modelagem das figuras vetoriais, o principal enfoque do SVG, é realizada através da composição de elementos que descrevem estruturas geométricas primitivas, como por exemplo: retângulos, círculos, elipses, linhas e polígonos. As composições podem ser agrupadas e transformadas conjuntamente através de elementos contêineres.

O SVG, de forma semelhante ao SWF, promove o reuso dos objetos gráficos através de estruturas de definição e instanciação. Essas estruturas, além de contribuírem para a criação de padrões gráficos, proporcionam a produção de arquivos menores.

Os objetos gráficos do SVG podem ser animados através de três abordagens: usando os elementos de animação do próprio vocabulário, usando o SVG DOM ou integrando o conteúdo com o SMIL.

O SVG possui os seguintes elementos de animação: *animate*, *set*, *animateMotion*, *animateColor* e *animateTransform*. Esses elementos descrevem as variações dos valores de atributos e propriedades dos elementos gráficos através do tempo. O controle do tempo é determinado por atributos específicos nos elementos de animação. Entre os atributos de controle de tempo, estão atributos que determinam o momento de início (*begin*), término (*end*) e duração (*dur*) da animação. Exceto pelo elemento *set*, que apenas utiliza um valor em sua definição, os demais elementos especificam as variações dos valores através de atributos que determinam valores chaves de origem (*from*) e destino (*to*), ou ainda, listas de valores (*values*) que devem ocupar ao longo da animação.

```
<svg width="8cm" height="3cm" viewBox="0 0 800 300"
  xmlns="http://www.w3.org/2000/svg">

  <rect x="10" y="100" width="0" height="50"
    fill="rgb(0,0,255)">
    <animate attributeName="width" attributeType="XML"
      begin="0s" dur="9s" fill="freeze" from="50" to="400" />
  </rect>
</svg>
```

Figura 3.2: Animação de um retângulo em SVG

¹² Um vocabulário XML é uma descrição de dados XML usados como meio de troca de informação, freqüentemente dentro de um domínio específico de atividade humana [AND 01]. Os termos: aplicativo XML e linguagem baseada em XML, também são empregados com esta mesma semântica [KIR 00, HOL 01].

¹³ O ECMAScript é a padronização do JavaScript pela ECMA.

¹⁴ O termo elemento, utilizado nesta seção, faz referência ao conceito de elemento empregado na nomenclatura do XML.

A figura 3.2 ilustra um documento em SVG que representa a animação de um retângulo azul aumentando de tamanho.

Além dos elementos de animação, os objetos gráficos podem ser animados utilizando linguagens de script. O SVG oferece um conjunto adicional de interfaces DOM para suportar animação via linguagens de script, o que possibilita linguagens como o ECMAScript iniciar e controlar animações. Esta abordagem é utilizada principalmente na construção de animações interativas.

3.3.3 Graphics Interchange Format (GIF)

O GIF, desenvolvido pela CompuServe Incorporate, é um protocolo para intercâmbio de imagens mapa de bits. É um dos protocolos mais utilizados para animações curtas na Internet, formadas por seqüência de quadros. A versão 89a desse protocolo, chamada popularmente de GIF animado, é amplamente utilizada na Web para criação de pequenas animações. Entre as principais aplicações, está a criação de animações para propagandas on-line e ícones animados.

Ao contrário do SWF e SVG, o GIF armazena as imagens resultantes do processo de animação. Embora seja mais aplicado em animações bidimensionais na Web, o GIF também pode armazenar imagens decorrentes de sistemas de animação tridimensionais ou até mesmo vídeo sem som.

Há diversos sistemas de animação dedicados a construção de GIF animados. Em geral, esses sistemas proporcionam a produção da animação a partir da criação de cada quadro da seqüência animada. Por exemplo, o Corel PHOTO-PAINT e o Adobe ImageReady possuem um ambiente de trabalho com ferramentas de desenho e efeito baseadas em processamento de imagem mapa de bits, e disponibilizam uma área de trabalho organizada para a criação de cada quadro. Esse processo de criação, embora proporcione uma grande liberdade de desenho ao autor, uma vez concluído, não contribui com o reuso de partes da animação nem com a manutenção da seqüência animada.

A figura 3.3 ilustra a estrutura do GIF, a qual é formada a partir de blocos e sub-blocos dedicados a armazenar informações de controle, gráficas e de propósito geral. Todo arquivo GIF possui um cabeçalho de identificação, um descritor de tela com os parâmetros necessários para definir os recursos de visualização, e é finalizado com um bloco terminador. As tabelas de cor utilizadas como referência pelas imagens, tanto global quanto local, são opcionais. Em geral, para manter o arquivo menor, as imagens contidas na animação fazem referência apenas à tabela de cor global, eliminando assim, as tabelas locais. Entre a tabela de cor global e o bloco finalizador podem existir vários blocos gráficos ou de propósito geral.

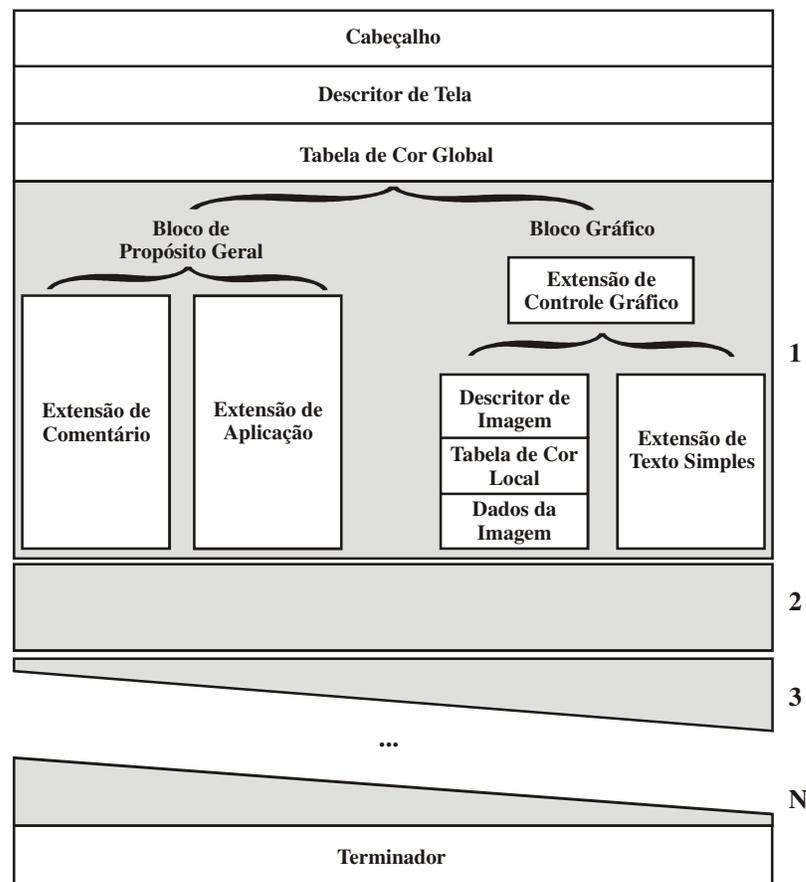


Figura 3.3: Estrutura do GIF versão 89a

Os blocos gráficos são utilizados para armazenar as informações relativas aos quadros da animação ou textos¹⁵. A extensão de controle gráfico possui informações quanto ao tempo de permanência do próximo elemento gráfico, e também indica a maneira como este deve ser tratado¹⁶ após ter sido mostrado.

3.3.4 Moving Picture Expert Group (MPEG-1), QuickTime e Audio Video Interleave (AVI)

Os formatos MPEG-1, QuickTime e AVI foram desenvolvidos com o objetivo de codificar vídeo e áudio associados para armazenamento em mídia digital. Esses formatos têm sido utilizados na Web para o intercâmbio de vídeos.

A codificação desses formatos está estruturada no armazenamento dos sucessivos quadros do filme destinados à apresentação em tempo real. Com esse objetivo, a codificação das imagens envolve algoritmos de compressão com perdas para a redução das redundâncias tanto espaciais quanto temporais, favorecendo a transferência em ambientes como a Web. A codificação com perdas baseia-se no conceito de comprometimento da precisão da imagem reconstruída em troca de uma maior compressão.

¹⁵ A utilização da extensão de texto simples é um recurso pouco utilizado para unir textos às imagens nas animações.

¹⁶ Os gráficos podem ser deixados no lugar ou removidos restaurando a cor de fundo ou a imagem anterior.

Embora esses formatos também possam ser aplicados à codificação de animações bidimensionais, a distorção nas imagens causadas pelos algoritmos de compressão com perda não tem estimulado está prática. Por exemplo, o MPEG-1 e QuickTime utilizam algoritmos para compressão espacial semelhantes ao JPEG, o qual distorce imagens que contenham bordas bem definidas [MAR 99].

3.4 Considerações Finais

Este capítulo traz um estudo sobre as animações por computador, a fim de que este assunto seja entendido para que se possa dar continuidade a dissertação. Vários formatos de animação são encontrados cada um com um fim específico. O modelo AGA utiliza a idéia de quadro a quadro e uma animação bidimensional. É importante ressaltar que este modelo será utilizado para a aplicação das operações categorias abordadas nesta dissertação, por isto de fez necessário o estudo apresentado neste capítulo.

4 MODELO AGA

4.1 Introdução

O modelo AGA surgiu do trabalho de dissertação do aluno de Mestrado do PPGC da UFRGS, Fernando Accorsi. O modelo, na íntegra, está na biblioteca do PPGC da UFRGS com o nome de **Animação Bidimensional para World Wide Web Baseada em Autômatos Finitos**. [ACC 2002]

A implementação do Modelo foi efetuada pelo aluno Guilherme de Campos Magalhães como projeto de Diplomação cujo trabalho foi intitulado **Aga Player: Animação 2D Baseada em Autômatos para Web**. [MAH 2002]

O modelo AGA especifica as alterações dos atores na animação a partir de uma estrutura baseada em autômatos finitos com saída [MEN 2000a]. Cada um dos atores na animação é especificado a partir de uma extensão para o autômato com saída, a qual vincula sons ou 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 Figura 4.1, 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 $q4$.

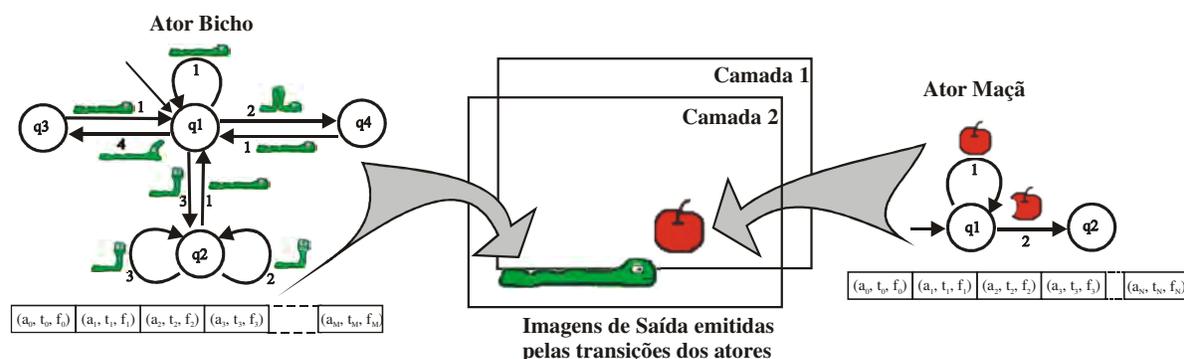


Figura 4.1: Atores especificados no modelo AGA

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 aplicada à imagem associada a 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.

4.2 Autômatos Finitos com Saída

As extensões propostas frente, a Máquina de Mealy [HOP 1979, MEN 2000a] serão representadas pela 6-upla, $ME = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$, onde:

Q conjunto finito de estados possíveis do autômato;

Σ alfabeto de símbolos de entrada;

Δ alfabeto de símbolos de saída;

δ função programa ou função de transição (função parcial $\delta: Q \times \Sigma \rightarrow Q$);

λ função de saída¹⁷ (função parcial $\lambda: Q \times \Sigma \rightarrow \Delta^*$);

q_0 estado inicial do autômato ($q_0 \in Q$).

4.3 Função de Saída Contextual

Conforme o modelo de Accorsi o alfabeto de saída Δ , para o ator AGA, é definido como um conjunto finito de imagens estáticas, onde cada elemento é uma imagem distinta que o ator pode utilizar durante a apresentação. O conjunto Δ^* , portanto, é composto por todas as imagens que podem ser produzidas pela composição de zero ou mais imagens de Δ .

A operação de composição pode ser definida da maneira mais conveniente à classe de animações representadas. Para a elaboração dos exemplos e a implementação do modelo, a palavra $i_1 i_2 \dots i_n \in \Delta^*$ representa a composição $i_1 + i_2 + \dots + i_n = i_r$ que produz a imagem i_r a partir da sobreposição da imagem i_1 pela i_2 e assim sucessivamente até i_n . Dependendo do instante da animação, a imagem de saída pode sofrer transformações com o objetivo de se adequar ao contexto da cena representada. Por exemplo, a mesma imagem pode aparecer em posições ou escalas diferentes. Para esse propósito, é introduzido o conjunto F , cujos elementos são funções do tipo $f(v \rightarrow, i) = i'$, onde i e i' 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' . O conjunto Δ^F é composto por todas as imagens de Δ^* transformadas por zero ou mais funções de F .

Assim, a função de saída estendida, chamada de função de saída contextual, é definida como a função parcial $\lambda_c: Q \times \Sigma \times F^* \rightarrow \Delta^F$, onde F^* é o conjunto de cadeias $\varphi = f_1 f_2 \dots f_m$ formadas por zero ou mais funções de F . Logo $\lambda_c(q, a, \varphi) = f_1(v_1 \rightarrow, f_2(v_2 \rightarrow, \dots f_m(v_m \rightarrow, i)))$, onde a imagem $i \in \Delta^*$ está associada a transição a que diverge do estado 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. Por exemplo, para o ator *bicho* apresentado na figura 4.1, o conjunto F possui a função *Mirror*, a qual produz o

¹⁷ Em [HOP 1979], o modelo definido para a Máquina de Mealy grava apenas um símbolo a cada transição, já em [MEN 2000a], a definição promove a associação de uma palavra de saída a cada transição. Para esse trabalho, a segunda definição foi utilizada por ser mais oportuna para a extensão do modelo.

espelhamento da imagem no seu eixo horizontal (argumento H) ou vertical (argumento V). Então, para a animação representada na figura 4.2, a leitura da terceira célula da fita de entrada $(1, \dots, Mirror(V))$ provoca a transição do estado q_4 para o estado q_1 e permuta a imagem 2 corrente pela imagem 1 transformada pela função $Mirror(V)$.

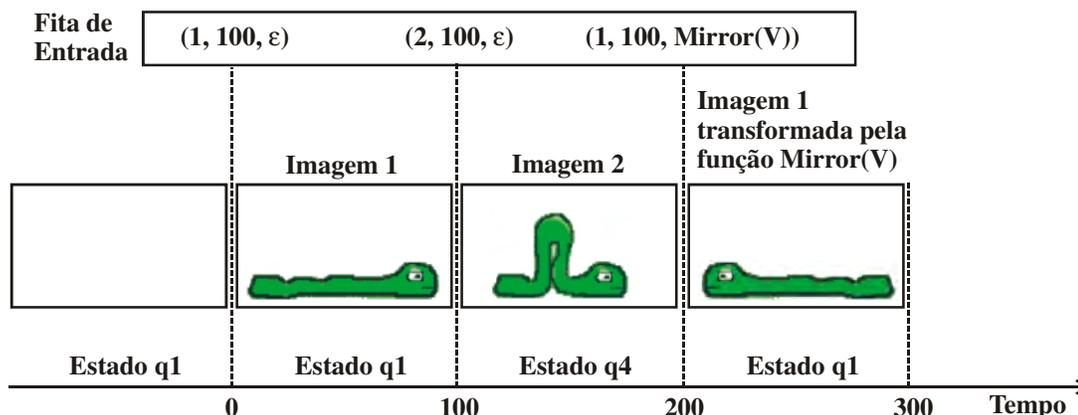


Figura 4.2: Troca de imagens do ator bicho provocadas pela leitura da fita

4.4 Função de Transição Temporal

No modelo AGA, além de determinar quais as variações que os atores podem sofrer durante a animação, é necessário especificar em que momentos deverão ocorrer. Como as variações estão associadas às transições do autômato, é preciso determinar em que momentos essas variações deverão ocorrer durante o processo de avaliação da fita de entrada.

Com este propósito, cada símbolo da entrada é associado a um número natural que determina o tempo de espera, em milésimos de segundos, após a leitura de um símbolo. Desta maneira, a fita de entrada $ft=(a_1, t_1, \varphi_1)(a_2, t_2, \varphi_2) \dots (a_n, t_n, \varphi_n)$ possui os triplos ordenados $(a, t, \varphi) \in \Sigma \times N \times F^*$, 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 da saída.

Para o comportamento temporal do autômato, as transições são consideradas instantâneas e o tempo passa durante a ocupação de um estado. Assim, para a entrada $(a_1, t_1, \varphi_1)(a_2, t_2, \varphi_2) \dots (a_n, t_n, \varphi_n)$ que provoca a seqüência de ocupação dos estados $q_0 q_1 \dots q_n$, o tempo t_k , para algum $1 \leq k \leq n$, determina o tempo de ocupação do estado q_k .

Deste modo, a função de transição é estendida para comportar um argumento adicional que determina o tempo que deve ser esperado após ser efetuada a transição. Esse argumento é obtido a cada célula lida a partir da fita de entrada. Portanto, a função de transição temporal é definida como a função parcial $\delta t: Q \times \Sigma \times N \rightarrow Q$, 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.

A figura 4.2 ilustra uma seqüência animada com 300 milésimos de segundo de duração. Inicialmente, o ator AGA se encontra no estado q_1 e no instante 0, com a leitura da primeira célula $(1, 100, \epsilon)$, ocorre uma transição instantânea para o estado q_1 , projetando a imagem

vinculada a este arco como saída. Após essa transição é esperado um tempo de 100 milésimos de segundo para a leitura da próxima célula. Note que a imagem emitida como saída permanece visível até a próxima transição.

4.5 Animação como um Conjunto de Atores no modelo AGA

Reunindo as extensões propostas, o ator AGA é definido então como uma 9- 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;

δt função de transição temporal (função parcial $\delta t: Q \times \Sigma \times N \rightarrow Q$);

λc função de saída contextual (função parcial $\lambda c: Q \times \Sigma \times F^* \rightarrow F$);

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

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

F conjunto finito de funções de transformação de Δ^F ;

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 o triplo ordenado $(a, t, \varphi) \in \Sigma \times N \times F^*$, 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, sejam 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 figura 4.3 que ilustra a modelagem em AGA. A animação tem duração de 1900 ms 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* = $(\{q_1, q_2, q_3, q_4\}, \{1, 2, 3, 4\}, \Delta_1, \delta t_1, \lambda c_1, q_1, \sigma_1, \{Trans(x,y)\}, \{“Boca Aberta”, “Contraído”, “Alerta”\})$, onde $\Delta_1, \delta t_1$ e λc_1 estão descritos na figura 4.1 (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* = $(\{q_1, q_2\}, \{1, 2\}, \Delta_2, \delta t_2, \lambda c_2, q_1, \sigma_2, \{Trans(x,y)\}, \{“Mordida”\})$, onde $\Delta_2, \delta t_2$ e λc_3 estão descritos na figura 4.1 (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 $AGA= ((Bicho, ft_{Bicho}), (Maçã,ft_{Maçã}), \preceq)$, onde $Maçã \prec Bicho$. 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 1900 ms. Na figura 4.3, 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 figura 4.3, 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.

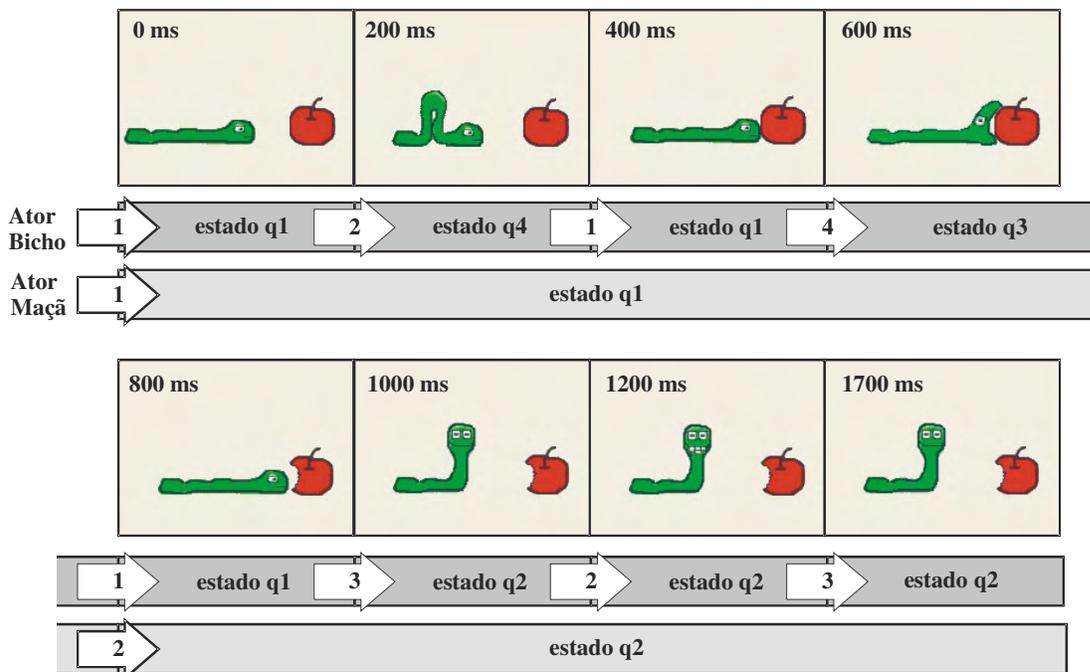


Figura 4.3: Animação envolvendo dois atores AGA

4.6 Considerações Finais

O modelo AGA é baseado em autômatos finitos determinísticos e possui características que favorecem sua aplicação na especificação e controle de animações para a *Web*.

O encapsulamento das propriedades estéticas e comportamentais dos atores em uma unidade básica, o ator AGA, favorece que estes sejam reutilizados em diferentes animações. As variações comportamentais destes atores em cada animação podem ser exploradas simplesmente pela troca da fita de entrada. Essas características também colaboram para que animações novas sejam formadas a partir da combinação de animações existentes.

No modelo AGA não é fundamentado matematicamente, como se pode 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.

Nesta dissertação se pretende estudar algumas operações matemáticas, utilizando Teoria das Categorias ao invés de Teoria dos Conjuntos pelas vantagens já mencionadas no Capítulo 1. E fundamentar matematicamente a obtenção de novas animações a partir dos atores originais, sem que se percam os autômatos originais e que se consiga reutilizá-los a qualquer momento quando se ache necessário.

Será apresentada uma extensão do modelo AGA, denominado AGANd (Animações Computacionais Baseadas em Autômatos Finitos Não determinísticos), sendo que o modelo AGA utiliza apenas autômatos finitos determinísticos.

5 AGAND (ANIMAÇÃO GRÁFICA BASEADA EM AUTÔMATOS FINITOS NÃO DETERMINÍSTICOS)

5.1 Introdução

Uma das maneiras de se representar sistemas regulares, ou seja, sistemas que podem ser expressos por um formalismo regular, é mapear o seu comportamento através de autômatos finitos.

Um autômato finito é definido como sendo um conjunto finito de estados, no qual um é identificado como estado inicial e um subconjunto de estados (eventualmente um conjunto vazio) é considerado como estado final. [AHO 1995, MEN 2000a, HOP 1979, DRO 1989].

Esses podem ser divididos em autômatos finitos determinístico (AFD) e autômatos finitos não determinístico (AFND).

Um autômato finito determinístico é aquele em que, no máximo, uma transição saindo de um estado pode ser possível para o mesmo símbolo de entrada. Um autômato finito não determinístico é aquele em que pode existir mais de uma transição saindo de um estado para o mesmo símbolo de entrada [AHO 95]. O não determinismo é uma importante generalização dos modelos de máquinas, sendo de fundamental importância no estudo da Teoria da Computação e da Teoria das Linguagens Formais [MEN 2000a].

Com um AFND, é possível representar alguns sistemas de maneira simples, o que muitas vezes não se consegue com um AFD.

O AGAND é uma extensão do modelo AGA para representar animações mais realistas e flexíveis, pois no modelo AGA todas as animações são modeladas com autômatos finitos determinísticos.

5.2 Visão Geral do modelo AGAND

No mundo em que vivemos as ações que tomamos na maioria das vezes são de forma não determinística. Quando pegamos uma caneta a maioria das vezes podemos pegá-la com a mão direita, mas eventualmente poderemos pegá-la com a mão esquerda, isto é uma situação de não determinismo.

No caso de um boneco caminhando, não é possível afirmar com qual das duas pernas (direita ou esquerda) o boneco iniciará a caminhada, a isto chamamos de não determinismo. O autômato que representa a caminhada do boneco poderá ser modelado a partir do modelo AGAND conforme ilustra a figura 5.1.

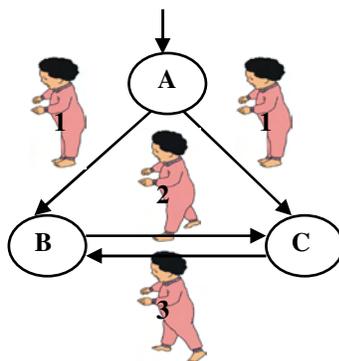


Figura 5.1: Animação modelada no AGANd

Tanto o modelo AGA quanto o modelo AGANd especificam a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação. As especificações em AGA, entretanto, são suportadas por um modelo formal baseado em AFD's e no AGANd as especificações são baseadas em AFND's.

5.3 Não determinismo interno e externo

Utilizando-se a semântica da Teoria da Concorrência [WIN 96], em que em uma situação de não determinismo um dos caminhos é escolhido, pode-se classificar os autômatos finitos não determinísticos por seu não determinismo em interno e externo. Entende-se por não determinismo interno aquele em que a máquina escolhe o caminho a seguir e não determinismo externo aquele em que um fator externo, por exemplo o usuário, escolhe o caminho a seguir.

Para exemplificar o não determinismo externo e interno pode-se tomar o caso de uma máquina de refrigerantes e seu usuário. A máquina de refrigerantes possui os seguintes botões: Guaraná (BOTÃO 1), Coca-cola (BOTÃO 2), Coca-cola (BOTÃO 3) e duas saídas, SAÍDA 1 e SAÍDA 2. O usuário pode pressionar qualquer botão para obter o seu refrigerante, sendo que se ele escolher a Coca-cola haverá duas opções, BOTÃO 2 e 3. A esta escolha feita pelo usuário, chamamos de não determinismo externo, em que algum fator externo, independente da máquina, decide o caminho a seguir. Por sua vez a máquina, de forma aleatória, escolherá a saída em que liberará o refrigerante. Nesse caso o usuário não tem nenhuma influência na escolha da máquina, ela é que define o caminho a seguir, a isso chamamos de não determinismo interno.

5.4 Utilização de pesos nas arestas do AFND - AFNDp

Uma sugestão é inserir peso nas arestas do AFND, para que este possa escolher quais dos caminhos deve seguir quando tiver um não determinismo em um de seus estados. Esta sugestão é bastante parecida com a encontrada pelos Autômatos Estocásticos e Redes de Marcov [KEM 60].

O autômato utilizado como base para definir o autômato finito não determinístico com pesos (AFNDp) é uma extensão dos autômatos finitos não determinísticos cuja a definição desse pode ser encontrado em [AHO 1995, MEN 2000, HOP 1979].

O autômato é representado em [MEN 2000a] como uma 5-upla: $M=(\Sigma, Q, \delta, q_0, F)$, onde Σ corresponde ao alfabeto de símbolos de entrada, Q o conjunto de estados possíveis do autômato o qual é finito, δ à função programa ou função de transição (função parcial $\delta: Q \times \Sigma \rightarrow 2^Q$); q_0 o estado inicial tal que q_0 é elemento de Q e F é o conjunto de estados finais, tal que F está contido em Q .

A extensão proposta no AFNDp diz respeito à função programa ou função de transição em que os arcos (transições) do autômato assumem pesos. Estes pesos são definidos pelo projetista da máquina.

A função de transição é denotada como $\delta^P(q, \alpha)$ e a sua definição estendida para $\delta^P: Q \times \Sigma \rightarrow 2^{(Q \times P)}$, em que P é o conjunto de valores dos pesos correspondentes a cada transição. P é o conjunto dos números Naturais menos o zero ($P=N^{-\{0\}}$).

O novo autômato ficaria: $M_p=(\Sigma, Q, \delta^P, q_0, F)$.

A figura 5.2 representa um ator modelo no AGANd utilizando um AFNDp para solucionar o não determinismo.

Quando no diagrama os pesos não forem definidos, por convenção é assumido o peso como 1.

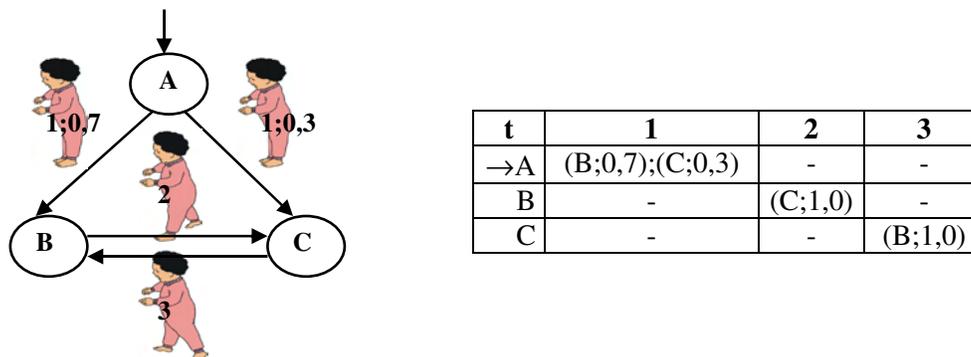


Figura 5.2: Utilização de um AFNDp no modelo AGANd. (A direita o AFNDp está representado na forma gráfica e a esquerda representado na forma tabular)

Na figura 5.2, no estado A com o símbolo de entrada 1 tem-se duas possibilidades de transição, a máquina escolhe um dos caminhos [WIN 96] a seguir, conforme o peso de suas transições. Será feito um cálculo de cada peso dividido pela soma dos pesos das transições. No estado A para o mesmo símbolo de entrada, neste caso 1, tem-se assim $7/10$ e $3/10$, isto é interpretado como sendo a probabilidade da máquina escolher em 70% dos casos a aresta com peso 7 e em 30% dos casos a outra aresta.

5.5 Considerações Finais

Utilizando-se, na construção de atores, o modelo AGANd, é possível compor animações de forma mais realística e mais flexíveis, conseguindo-se assim animações que tenham um comportamento mais próximo com o que ocorre no mundo real. O modelo AGA não havia ainda explorado esta idéia, que para modelar alguns tipos de animações se

torna mais viável pois muitas vezes a modelagem em AFND resulta em um autômato com um menor número de estados do que o AFD equivalente.

Uma das maneiras de se interpretar um AFND é com a utilização de um AFNDp, que se baseia nos princípios das Redes de Markov e dos Autômatos Estocásticos. O AFNDp aumenta o poder de expressão dos autômatos finitos não determinísticos utilizando a noção de pesos em suas arestas. No modelo proposto a máquina escolhe aleatoriamente o caminho a seguir a partir dos pesos especificados pelo projetista da mesma, que é responsável pela definição da prioridade de cada um dos caminhos.

6 APLICAÇÃO DE PRODUTO E COPRODUTO ENTRE ATORES DE UMA ANIMAÇÃO

6.1 Introdução

Grafos são freqüentemente usados para representar sistemas. Exemplos típicos são os formalismos Autômatos, Sistemas de Transição e Redes de Petri. Nesses casos, as construções categoriais de Produto e Coproduto podem ser interpretadas como combinadores de sistemas. [MEN 2002]

Este capítulo trata das construções categoriais de Produto e Coproduto, aplicando-os em animações computacionais baseadas em autômatos finitos. Os conceitos aqui explorados, de Produto e Coproduto são baseados em grafos, já que se pode dizer que um autômato é um grafo rotulado e direcionado ([HOP 69], [HOP 79], [AHO 95]).

6.2 Definição de Produto e Coproduto em Grafos

Faz-se necessário especificar o conceito categorial de grafos, sendo que serão utilizadas as operações categoriais de produto e coproduto e demais operações categoriais estudadas nessa dissertação, todas baseadas em grafos.

Um grafo \mathbf{G} é uma quádrupla: $\mathbf{G}=\langle V, T, \partial_0, \partial_1 \rangle$, na qual: V é um conjunto de nodos ou vértices; T é um conjunto de arcos; $\partial_0, \partial_1: T \rightarrow V$ 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 $\partial_1(t)=B$.

Pode-se dizer que Produto de 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.

Sejam dois grafos $\mathbf{G}_1=\langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $\mathbf{G}_2=\langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ o Produto dos dois grafos resulta em: $\mathbf{G}_1 \times \mathbf{G}_2 = \langle V_1 \times V_2, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle$. O coproduto de \mathbf{G}_1 e \mathbf{G}_2 resulta em: $\mathbf{G}_1 + \mathbf{G}_2 = \langle V_1 + V_2, T_1 + T_2, \partial_{01} + \partial_{02}, \partial_{11} + \partial_{12} \rangle$.

O objeto resultante do Produto é um grafo no qual nodos e arcos são pares dos nodos e arcos dos grafos componentes, como ilustra a figura 6.1.

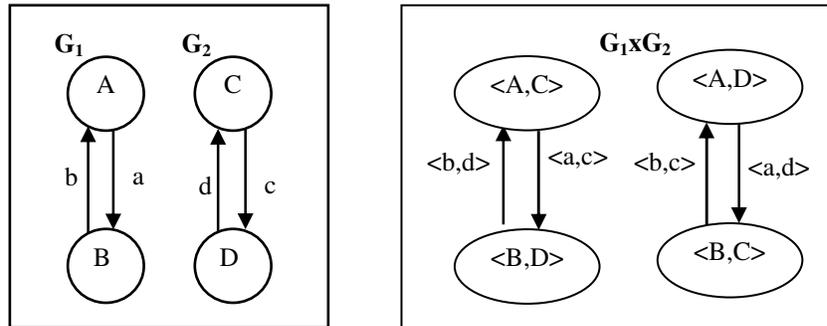


Figura 6.1: Produto de Grafos

O objeto resultante do Coproduto é um grafo distribuído, constituído pelos grafos G_1 e G_2 justapostos, lado a lado, como ilustrado na figura 6.2.

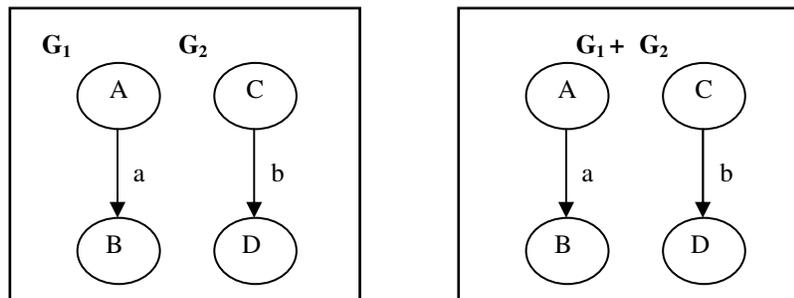


Figura 6.2: Coproduto de Grafos

Na representação do comportamento conjunto de modelos de sistemas baseados em grafos, Produto e do Coproduto são interpretados como combinadores síncrono e assíncrono, respectivamente. Entretanto, no caso de autômatos, o Coproduto caracteriza um não-determinismo ou escolha, no sentido em que cada um dos grafos componentes representa uma alternativa.

6.3 Aplicação do Produto

Tomando-se como base o modelo AGA ou o modelo AGAND, pode-se aplicar a operação categorial de produto entre os atores de uma animação. O objeto resultante é um novo ator formado pelos pares de nodos e arcos do ator original. Accorsi ([ACC 2000] e [ACC 2001]), utiliza o termo “atores” para identificar cada um dos autômatos (objetos) da animação, e uma animação é o conjunto formado pelos atores e a fita de entrada.

O estado inicial dos autômatos a serem utilizados nessa dissertação 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 animação. A figura 6.3 representa dois atores de uma animação.

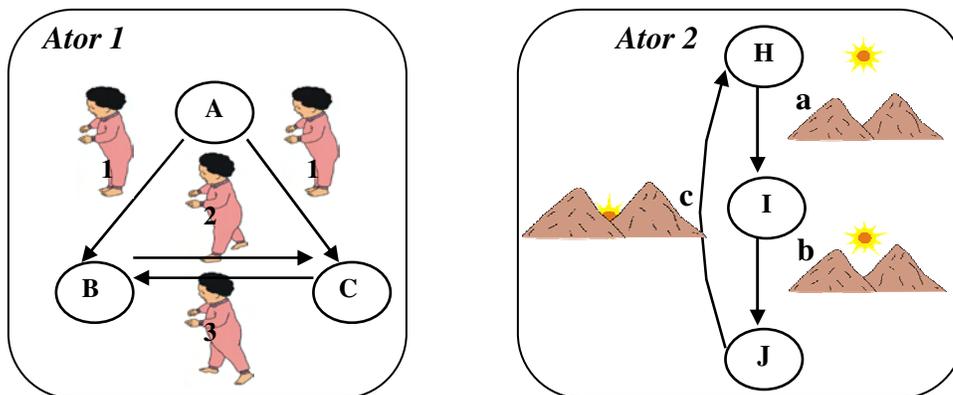


Figura 6.3: Atores de uma animação

A figura 6.4 ilustra o Produto aplicado entre os dois atores representados na figura 6.3. Pode-se observar que o objeto resultante são todas as combinações possíveis entre os estados dos atores originais. Este autômato está desenhado somente como uma forma didática para que se possa entender mais claramente a operação categorial em questão.

Todas os autômatos resultantes das operações categoriais que serão estudadas nesta dissertação serão desenhados somente para que se possa enxergar o resultado da operação de uma maneira mais clara, porém o resultado é obtido a partir da aplicação matemática da operação categorial em questão sobre os autômatos dos atores 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 categorial.

Accorsi [ACC 2002] utiliza a noção de Produto Categorial em suas animações, porém ele chama isto de composição, definido no capítulo 4. Esta composição gera uma nova animação a partir das animações originais, porém este controle é feito pela fita de entrada¹⁸. Para obter esta nova animação com sincronismo Accorsi trabalha com a célula da fita de entrada correspondente ao tempo. É feito um controle do tempo de cada frame para que se obtenha o resultado desejado.

Utilizando-se a operação categorial de produto são conseguidas todas as combinações possíveis dos atores originais e também o sincronismo entre os mesmos, sendo que para que isso ocorra não é necessário que se controle o tempo de cada frame, isto é obtido automaticamente com a aplicação da operação matemática de produto descrita no capítulo 6.2.

¹⁸ Cada ator da animação possui uma fita individual.

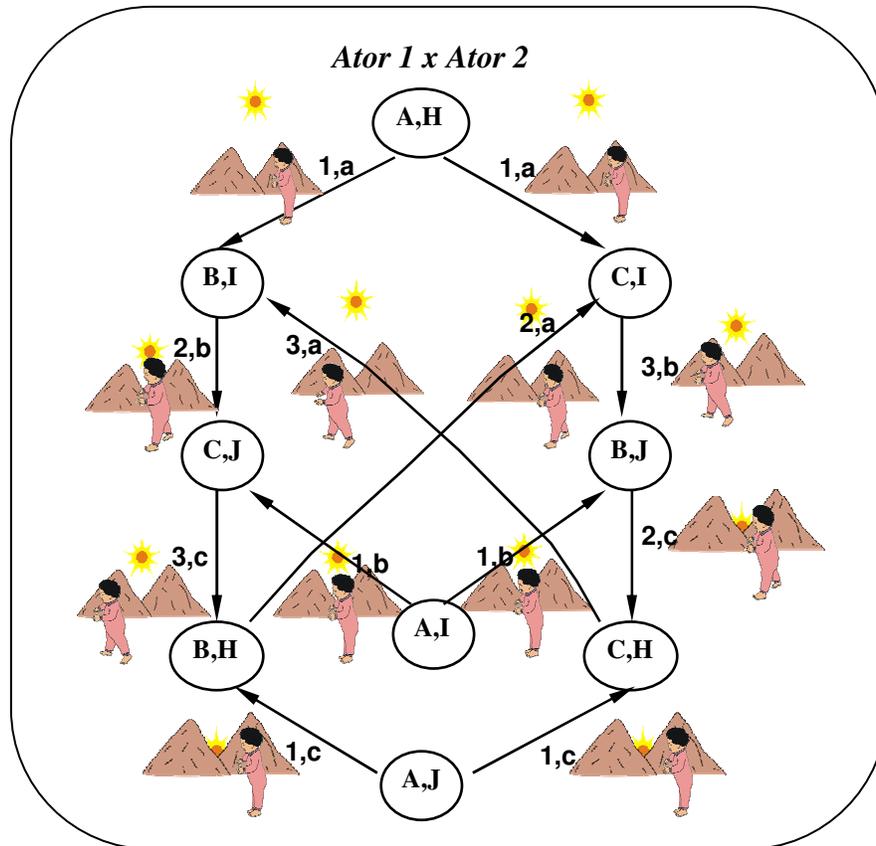


Figura 6.4: Produto entre atores de uma animação

6.3.1 Fita de entrada

Sendo que, nessa dissertação, a operação de produto é feita sobre os atores e não sobre a fita é necessário que uma nova fita de entrada seja definida para essa operação. A fita será definida da seguinte forma:

$$ft = (\langle a_1 ator^1, a_1 ator^2 \dots a_1 ator^n \rangle, \langle j_1 ator^1, j_1 ator^2 \dots j_1 ator^n \rangle, t_1), \\ (\langle a_2 ator^1, a_2 ator^2 \dots a_2 ator^n \rangle, \langle j_2 ator^1, j_2 ator^2 \dots j_2 ator^n \rangle, t_2) \dots \\ (\langle a_n ator^1, a_n ator^2 \dots a_n ator^n \rangle, \langle j_n ator^1, j_n ator^2 \dots j_n ator^n \rangle, t_n)$$

Na qual cada célula possui o os elementos ($\langle a ator^1, a ator^2 \dots a ator^n \rangle, \langle j ator^1, j ator^2 \dots j ator^n \rangle, t$) onde:

a → um símbolo do alfabeto de entrada

j → cadeia de funções de transformação da saída

t → tempo de transição entre os frames da animação

A figura 6.5 ilustra como ficaria a animação com a utilização desta fita única.

Fita de Entrada		
a	j	t
1	ϵ	100
a	ϵ	
3	ϵ	100
b	ϵ	
2	ϵ	100
c	ϵ	

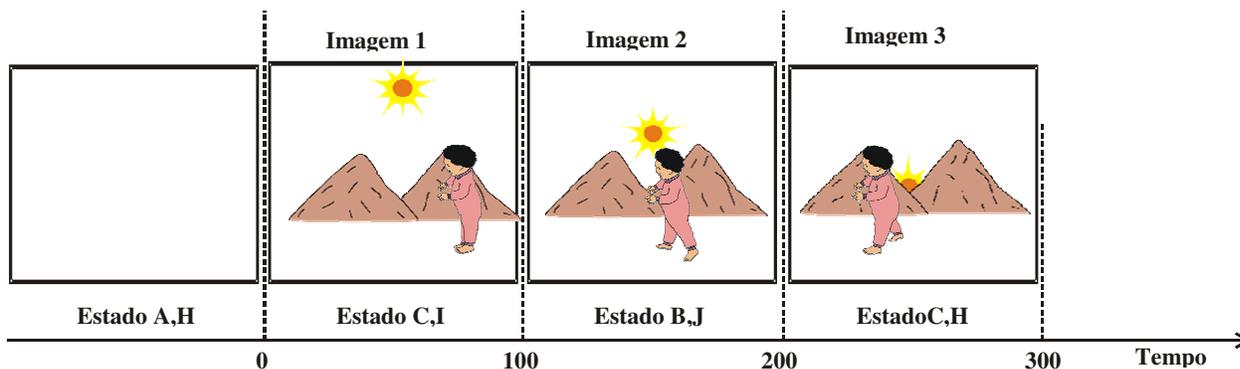


Figura 6.5: Fita de Entrada única para o Produto entre atores de uma animação

Conforme definido por Accorsi a área de atuação dos atores continua sendo dividida em camadas onde as imagens são sobrepostas com o objetivo de estabelecer a ordem de sobreposição das imagens. Na figura 6.5 a imagem do *Ator 1* deve ser sobreposta a imagem do *Ator 2*. Assim, a imagem correspondente a cada instante da animação é formada pela unificação das camadas. Além disso o posicionamento e a transformação dos atores na área de animação é definido pelo segundo elemento da fita de entrada (**j**) que é o conjunto de transformações aplicadas à imagem associada a saída, no exemplo apresentado na figura 6.5 a imagem de saída dos atores não sofreu nenhuma transformação, por isso aparece o símbolo ϵ na fita de entrada.

Uma sugestão de implementação da operação categorial do Produto será apresentada no Capítulo 8.

6.4 Aplicação do Coproduto

A operação categorial de Coproduto resulta em um grafo distribuído, constituído pelos grafos originais justapostos lado a lado. É interpretado como combinador assíncrono de um sistema. O Coproduto caracteriza um não determinismo ou escolha, no sentido em que cada um dos grafos componentes do objeto resultante do coproduto representa uma alternativa [MEN 2002].

Pode-se especializar a noção de Coproduto e aplicá-lo entre atores de uma animação, tomando-se como base o modelo AGA ou o modelo AGANd. O objeto resultante é um novo ator, formado pelos atores originais justapostos lado a lado, em que um dos atores originais será apresentado na área de animação. A escolha de qual ator será apresentado é aleatória, ou seja, ela pode ser feita pelo usuário ou pela própria máquina. Os outros atores

envolvidos na operação de coproduto poderão ou não, ser apresentados posteriormente na área de animação. Novamente a escolha da seqüência de apresentação dos atores a ser exibida é aleatória.

É importante ressaltar que o estado inicial dos autômatos é indefinido conforme citado anteriormente no capítulo 6.3.

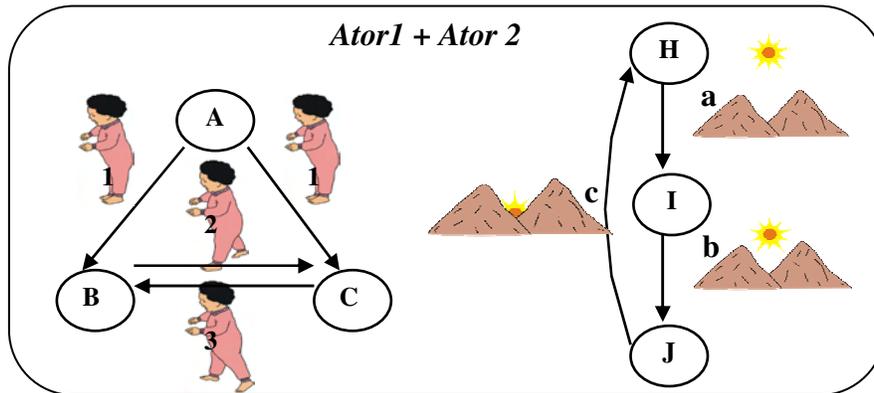


Figura 6.6: Coproduto entre atores de uma animação

O autômato resultante da operação categorial de coproduto (figura 6.6) entre os atores ilustrados na figura 6.3 é obtido através da operação matemática do coproduto explicada no Capítulo 6.2.

6.4.1 Fita de entrada

Sendo que nessa dissertação a operação categorial de Coproduto é aplicada sobre os atores e não sobre a fita de entrada, torna-se necessário que uma nova fita seja definida.

A fita de entrada, para a operação categorial de Coproduto, é definida como segue:

$$ft = \langle \langle a_1 ator^1, a_1 ator^2 \dots a_1 ator^n \rangle, \langle j_1 ator^1, j_1 ator^2 \dots j_1 ator^n \rangle, t_1 \rangle, \\ \langle \langle a_2 ator^1, a_2 ator^2 \dots a_2 ator^n \rangle, \langle j_2 ator^1, j_2 ator^2 \dots j_2 ator^n \rangle, t_2 \rangle \dots \\ \langle \langle a_n ator^1, a_n ator^2 \dots a_n ator^n \rangle, \langle j_n ator^1, j_n ator^2 \dots j_n ator^n \rangle, t_n \rangle$$

Na qual cada célula possui o os elementos $\langle \langle a ator^1, a ator^2 \dots a ator^n \rangle, \langle j ator^1, j ator^2 \dots j ator^n \rangle, t \rangle$ onde:

a → um símbolo do alfabeto de entrada

j → cadeia de funções de transformação da saída

t → tempo de transição entre os frames da animação

A figura 6.7 ilustra como ficaria a animação com a utilização dessa fita:

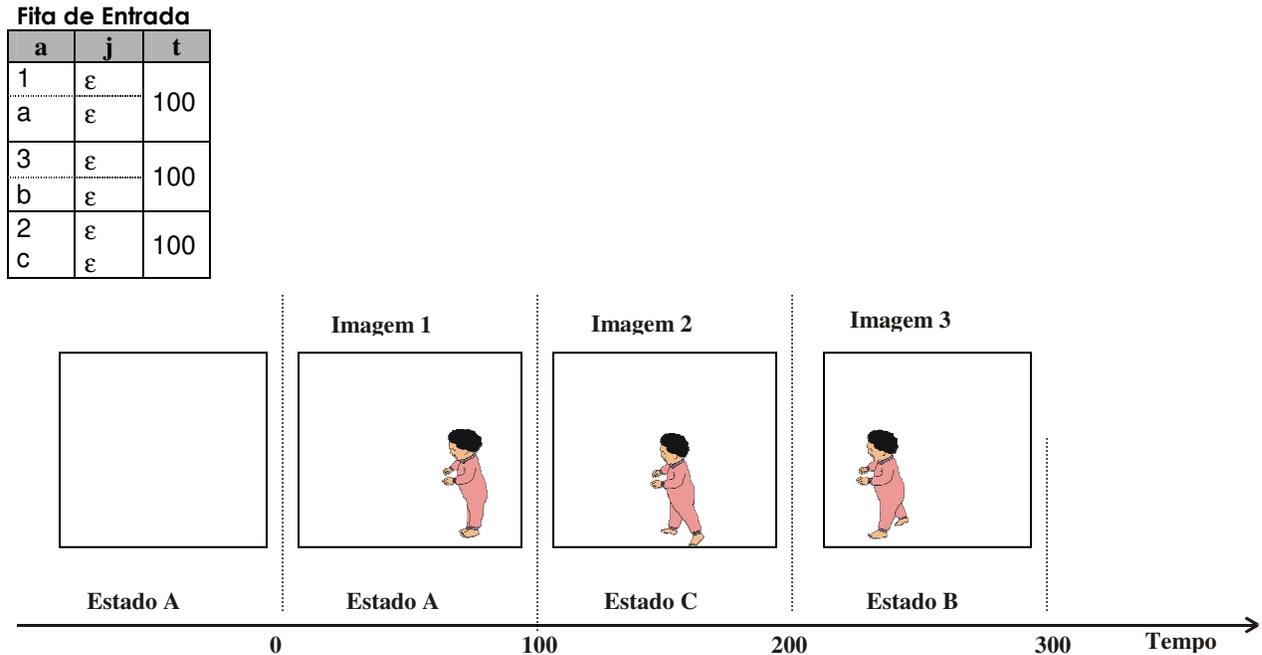


Figura 6.7: Fita de Entrada para o Coproduto

No capítulo 8 é apresentada uma sugestão de implementação para o Coproduto.

6.4.2 Comparação com o modelo AGA

No modelo AGA o mesmo resultado do Coproduto é obtido quando se entra com a fita de um só ator que compõe a animação ou se define a seqüência dos atores que serão apresentados na área de animação. A figura 6.8 mostra o autômato de dois atores no modelo AGA. Para a obtenção do mesmo resultado do coproduto de categorias Accorsi utiliza apenas uma fita e identifica de que ator é esta fita.

Já no Coproduto de Categorias entra-se com uma fita e aleatoriamente é escolhido um dos atores que deverá ser apresentado na tela de exibição.

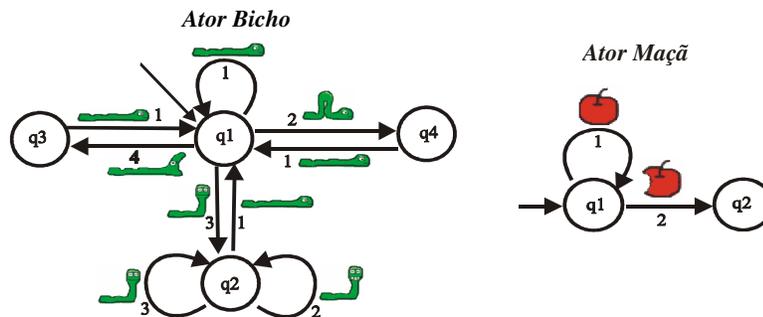


Figura 6.8: Atores do modelo AGA

Um exemplo de fita para o modelo AGA seria a definida abaixo:

$ft_{\text{Bicho}} = (1, 200, \text{Trans}(32,0)) (2, 200, \text{Trans}(32,0)) (1, 200, \text{Trans}(32,0)) (4, 200, \text{Trans}(32,0)) (1, 200, \text{Trans}(32,0)) (3, 200, \text{Trans}(32,0)) (2, 500, \text{Trans}(32,0)) (3, 200, \text{Trans}(32,0))$.

A fita já foi identificada como sendo do ator bicho. A animação obtida com a leitura dessa fita está representada na figura 6.9.

Já com a utilização do Coproduto de Teoria das Categorias a fita de entrada seria a seguinte:

$ft = (<1, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<2, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<1, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<4, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<1, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<3, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<2, 1>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200),$
 $(<3, 2>, <\text{Trans}(32,0), \text{Trans}(45,2)>, 200)$

A animação obtida poderia ser a mesma do modelo AGA figura 6.9 ou poderia ser a animação do Ator Maçã. Ou ainda a animação de um dos dois atores em primeiro seguido da animação do outro ator. Na figura 6.9, como exemplo, foi exibida a animação do ator bicho.

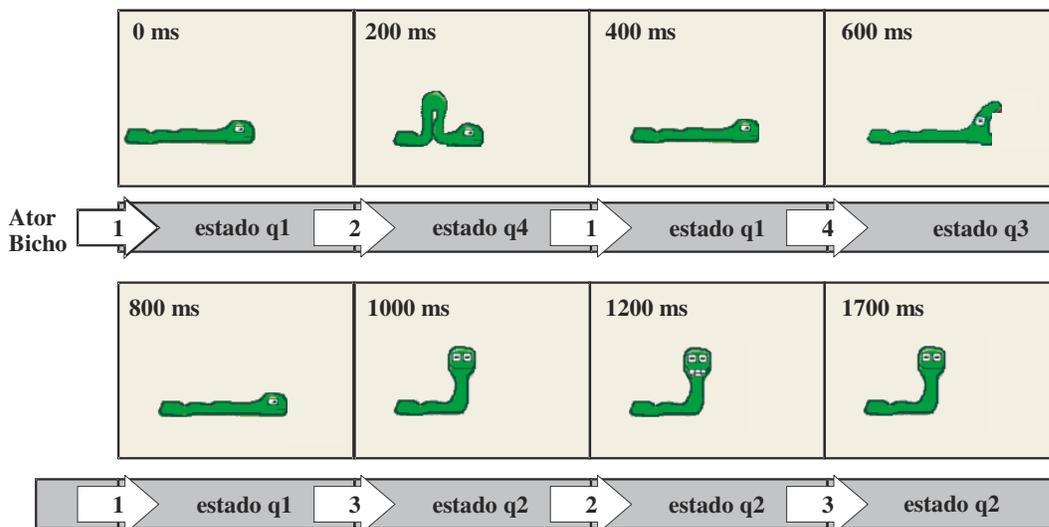


Figura 6.9: Animação

6.5 Produto e Coproduto de Grafos Reflexivos

Relativamente à categoria dos grafos reflexivos, e comparando com a dos grafos, tem-se que 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 sistemas componentes, conforme ilustra a figura 6.10.

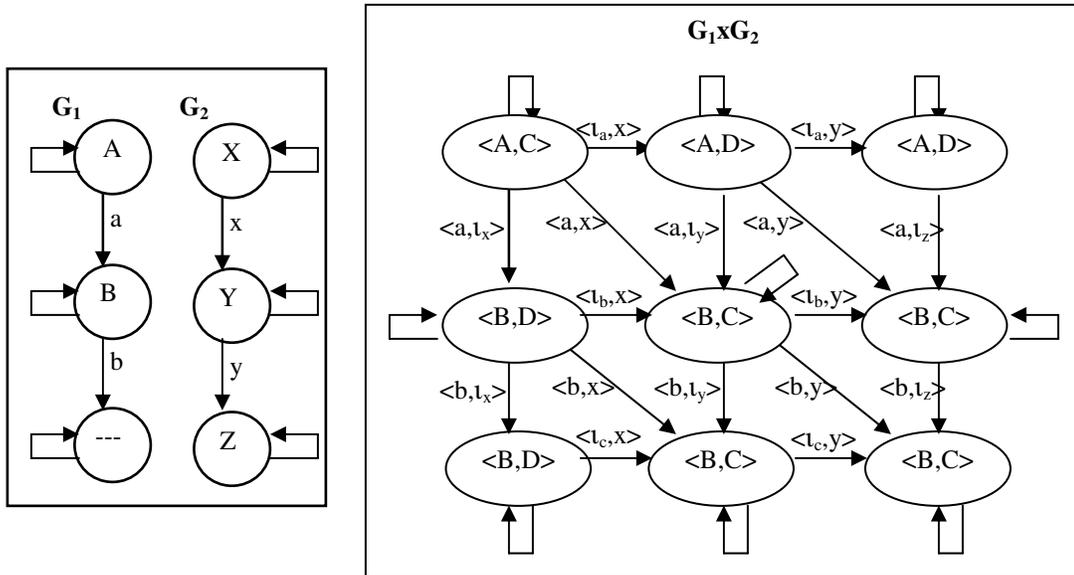


Figura 6.10: Produto de Grafos Reflexivos

O produto entre os grafos reflexivos G_1 e G_2 é definido pelo seguinte grafo:

$G_1 \times G_2 = \langle V_1 \times V_2, T_1 \times T_2, \partial_{0_1} \times \partial_{0_2}, \partial_{1_1} \times \partial_{1_2}, \iota_1 \times \iota_2 \rangle$, onde $\partial_{k_1} \times \partial_{k_2}$, para $k \in \{0, 1\}$ e onde $\iota_1 \times \iota_2$ é induzido pelo produto dos nodos e arcos como ilustrado na figura 6.11, juntamente com os morfismos $\pi_1 = \langle \pi_{V_1} \times \pi_{T_1} \rangle: G_1 \times G_2 \rightarrow G_1$ $\pi_2 = \langle \pi_{V_2} \times \pi_{T_2} \rangle: G_1 \times G_2 \rightarrow G_2$ onde π_{V_1} , π_{V_2} , π_{T_1} e π_{T_2} são como na categoria de grafos.

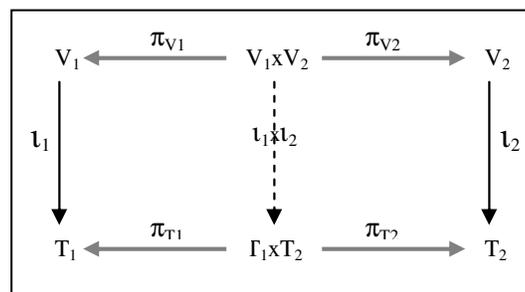


Figura 6.11: Diagrama comutativo adicional para produto de grafos reflexivos

No coproduto o objeto resultante pode ser visto como a justaposição, lado a lado dos grafos componentes como na categoria de grafos, conforme ilustra a figura 6.12. A sua fundamentação matemática está explicada no capítulo 6.2.

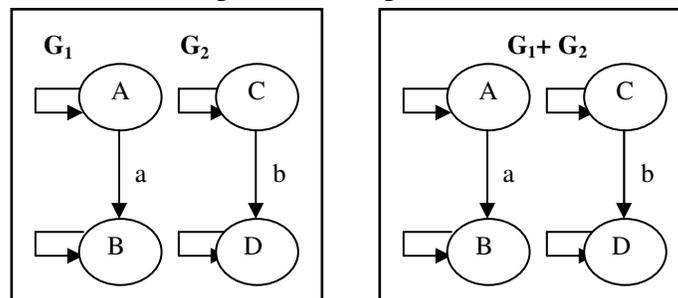


Figura 6.12: Coproduto de Grafos Reflexivos

O coproduto entre dois grafos reflexivos resulta no grafo $\mathbf{G}_1+\mathbf{G}_2=\langle V_1+V_2, T_1+T_2, \partial_{0_1}+\partial_{0_2}, \partial_{1_1}+\partial_{1_2}, \iota_1+\iota_2 \rangle$, onde $\partial_{k_1} \times \partial_{k_2}$, para $k \in \{0,1\}$, é como na categoria de grafos e onde $\iota_1+\iota_2$ é induzido pelo coproduto dos nodos e arcos como ilustrado na figura 6.13, juntamente com os morfismos $q_1=\langle q_{V_1}+q_{T_1} \rangle: G_1 \rightarrow G_1+G_2$ $q_2=\langle q_{V_2}+q_{T_2} \rangle: G_2 \rightarrow G_1+G_2$ onde q_{V_1} , q_{V_2} , q_{T_1} e q_{T_2} são como na categoria de grafos.

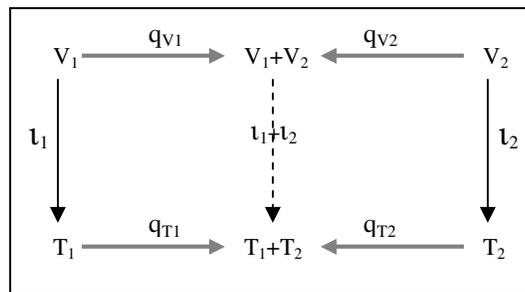


Figura 6.13: Diagrama comutativo adicional para coproduto de grafos reflexivos

6.5.1 Aplicação do Produto de Grafos Reflexivos

Tomando-se como base o modelo AGA ou o modelo AGANd, pode-se aplicar a operação categorial de produto de Grafos reflexivos entre os atores de uma animação, desde que estes sejam grafos reflexivos. O objeto resultante é um novo ator formado pelos pares de nodos e arcos dos atores originais.

É interessante ressaltar que o estado inicial dos autômatos a serem utilizados nessa dissertação 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 animação.

A figura 6.15 ilustra o produto de grafos reflexivos entre dois atores de uma animação representados na figura 6.14. Para simplificar, a figura do produto de grafos reflexivos está sem o desenho das saídas.

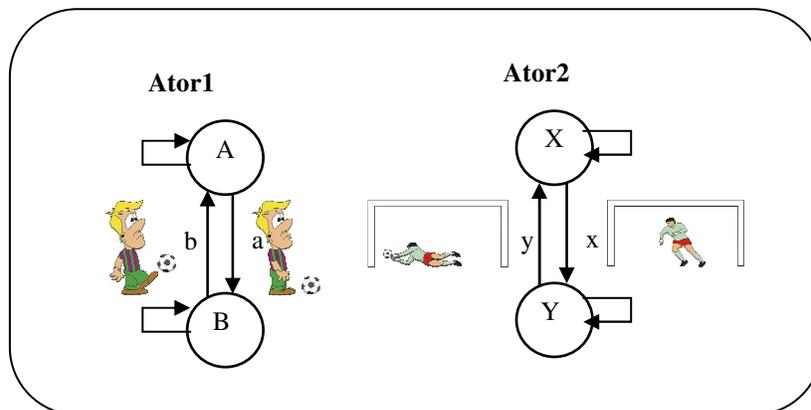


Figura 6.14: Atores de uma animação

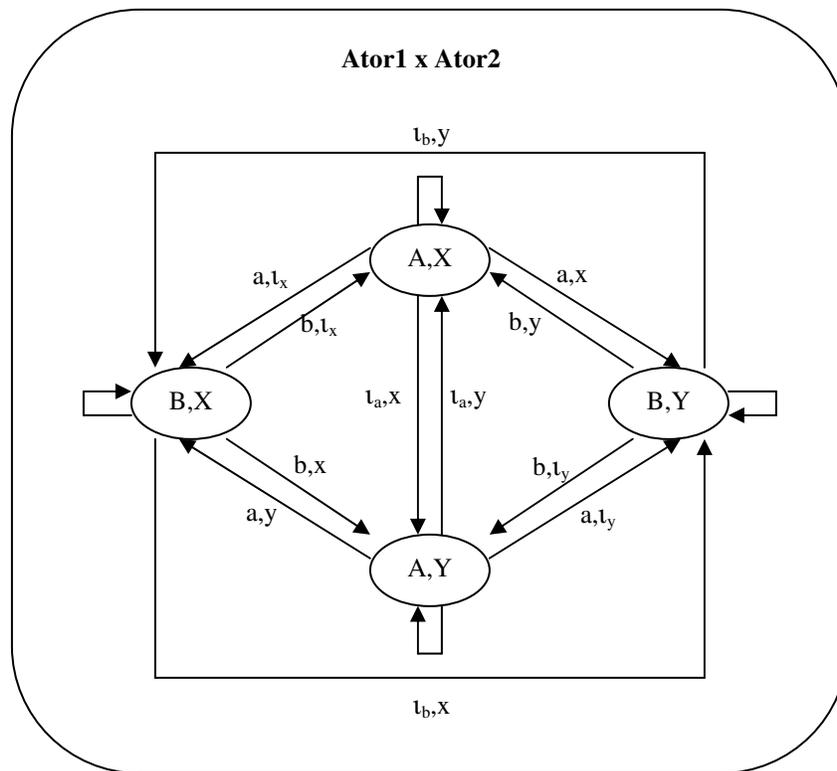


Figura 6.15: Produto de grafos reflexivos entre dois atores

O produto pode ser interpretado como um combinador simultaneamente síncrono e assíncrono.

6.5.2 Aplicação do Coproduto de Grafos Reflexivos

A operação categorial de Coproduto resulta em um grafo reflexivo distribuído, constituído pelos grafos reflexivos originais justapostos lado a lado. [MEN 2002] O Coproduto de grafos reflexivos é igual ao coproduto de grafos.

Pode-se especializar a noção de Coproduto e aplicá-lo entre atores de uma animação, tomando-se como base o modelo AGA ou o modelo AGANd. O objeto resultante é um novo ator, formado pelos atores originais justapostos lado a lado, em que um dos atores originais será apresentado na área de animação. A escolha de qual ator será apresentado é aleatória, ou seja, ela pode ser feita pelo usuário ou pela própria máquina. Os outros atores envolvidos na operação de coproduto poderão ou não, ser apresentados posteriormente na área de animação. Novamente a escolha da seqüência de apresentação dos atores a ser exibida é aleatória.

É importante ressaltar que o estado inicial dos autômatos é indefinido conforme citado anteriormente no capítulo 6.3.

A figura 6.16 ilustra a aplicação do Coproduto entre dois atores de uma animação representados na figura 6.15.

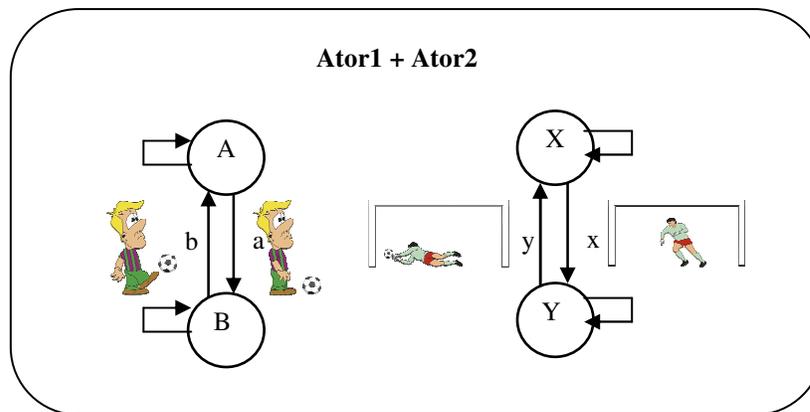


Figura 6.16: Coproduto de grafos reflexivos entre dois atores

6.6 Composição de Operações

Em Teoria das Categorias é possível que se aplique operações categoriais sobre operações categoriais, sejam elas quais forem e quantas vezes se ache necessário para que se obtenha o resultado desejado. A figura 6.18 ilustra a aplicação da operação categorial de Coproduto seguida da operação categorial de Produto. A figura 6.17 representa os atores envolvidos na operação.

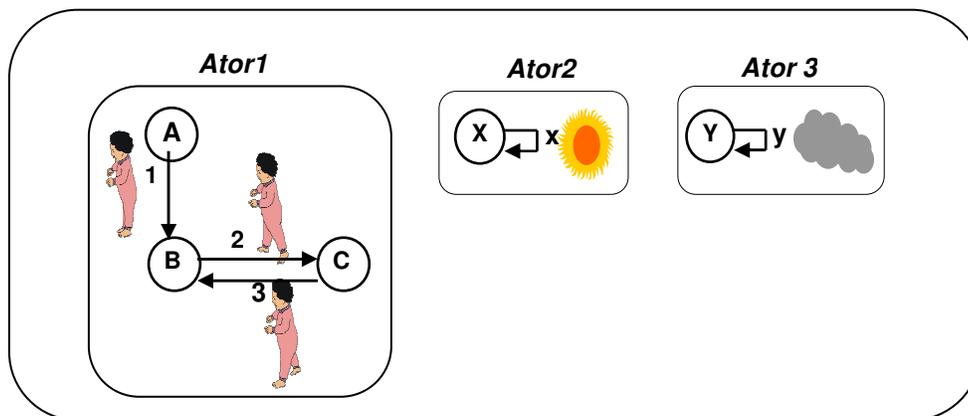


Figura 6.17: Atores de uma animação

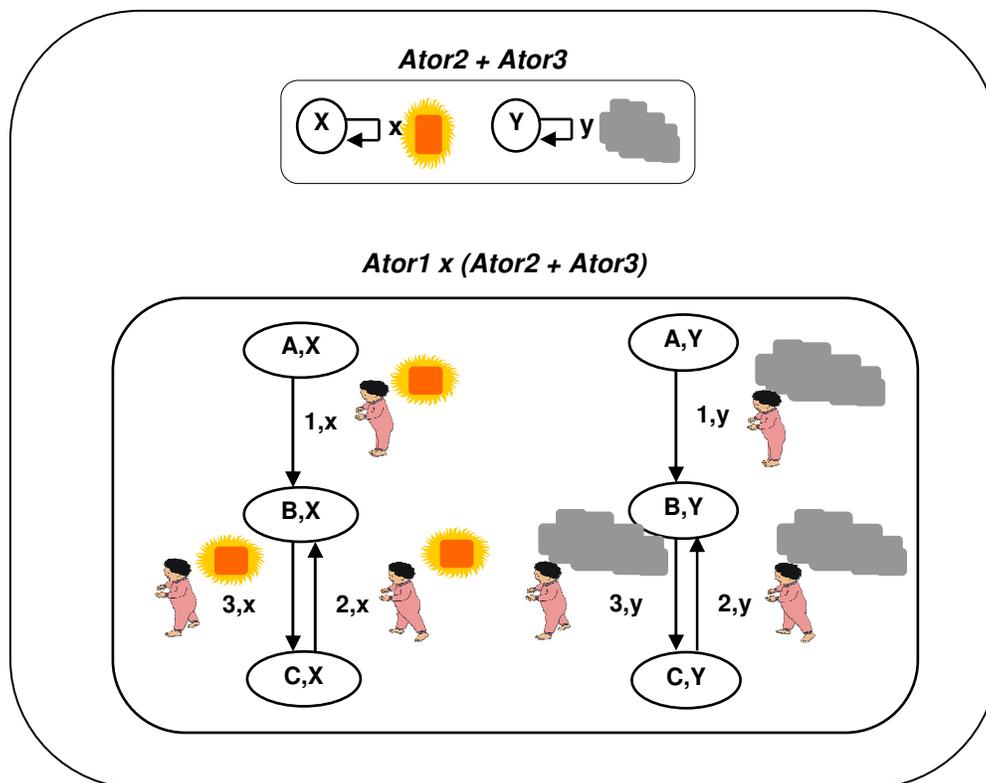


Figura 6.18: Produto entre o Ator1 e o resultado do Coproduto entre os Atores2 e 3

Outras operações poderiam ser aplicadas em cima das operações já feitas, tudo depende do resultado que se pretende obter. Nessa dissertação serão apresentados apenas alguns exemplos, que podem e devem ser explorados e ampliados pelo criador da animação.

6.7 Considerações Finais e Trabalhos Futuros

Neste capítulo se estudou as operações categoriais de Produto e Coproduto aplicadas em grafos. As animações computacionais serviram para verificar o resultado prático da aplicação dessas operações.

Pôde-se observar que tanto no Produto como no Coproduto o objeto resultante é sempre uma novo ator a partir dos atores originais, portanto se o autômato original for modificado o resultado das operações será alterado automaticamente.

O texto mostra ainda que é possível aplicar operações categoriais sobre operações categoriais quantas vezes se ache necessário. Como exemplo foi mostrado apenas o resultado da operação categorial de Coproduto seguida de Produto, mas no decorrer deste trabalho será visto que é possível englobar quaisquer operações categoriais que se desejar.

O texto não trata da aplicação das operações categoriais sobre as fitas de entrada das animações, este é um tema que poderá ser explorado em um trabalho futuro.

O objetivo do presente capítulo não é provar que a operação categorial de produto e coproduto é melhor ou pior do que outras maneiras para se compor animações. O objetivo dessa dissertação é fundamentar matematicamente as composições possíveis entre os atores para que se consiga uma nova animação a partir dos autômatos dos atores originais.

7 APLICAÇÃO DE PRODUTO FIBRADO E SOMA AMALGAMADA ENTRE ATORES DE UMA ANIMAÇÃO

7.1 Introdução

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.

Uma das interpretações de Soma Amalgamada consiste na composição de sistemas, identificando partes em comum. Tratando-se, portanto, de uma generalização da noção de união de Teoria dos Conjuntos.

Este capítulo trata da aplicação dessas duas operações categoriais em animações computacionais baseadas em autômatos finitos, sejam eles determinísticos ou não determinísticos.

O conceito de Produto Fibrado e Soma Amalgamada é apresentado com ênfase em grafos. O resultado da aplicação dessas operações categoriais em animações é apresentado nos capítulos seguintes.

7.2 Definição de Produto Fibrado e Soma Amalgamada em Grafos

Faz-se necessário especificar o conceito categorial de grafos, sendo que serão utilizadas as operações categoriais de produto fibrado e soma amalgamada baseadas em grafos.

Um grafo G é uma quádrupla: $G = \langle V, T, \partial_0, \partial_1 \rangle$, na qual: V é um conjunto de nodos ou vértices; T é um conjunto de arcos; $\partial_0, \partial_1: T \rightarrow V$ 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 $\partial_1(t) = B$.

Pode-se dizer que Produto Fibrado é um limite de um diagrama constituído por dois morfismos com um mesmo objeto destino. Soma Amalgamada é o seu conceito dual, ou seja, um colimite de um diagrama constituído por dois morfismos com um mesmo objeto origem. Portanto, generalizam as noções de equalizador e co-equalizador respectivamente. [MEN 2002]

Pode-se calcular um Produto Fibrado de grafos utilizando o cálculo do Produto Fibrado usando Produto e Equalizador [MEN 2002, pág.193]. Uma descrição do cálculo de Produto Fibrado de grafos é explicada de forma mais ditática em Menezes [2002, pág. 200], teorema 6.20 – Produto Fibrado em Gr.

Para a interpretação que foi dada ao Produto Fibrado de grafos como restrição de sistemas, em que uma parte do sistema original é mantida preservando as suas

características, é apresentada na figura 7.1. Essa abordagem pode ser usada, por exemplo, em Reuso ou Herança de “código”, ou seja, o aproveitamento (reuso) de parte de um sistema existente, como em alguns tipos de herança na programação orientada a objetos.

Uma outra interpretação para Produto Fibrado seria a utilização do mesmo como sincronização de sistemas de acordo com uma tabela específica de sincronização. No livro do Menezes [2002, pág 202] é apresentado um exemplo dessa sincronização.

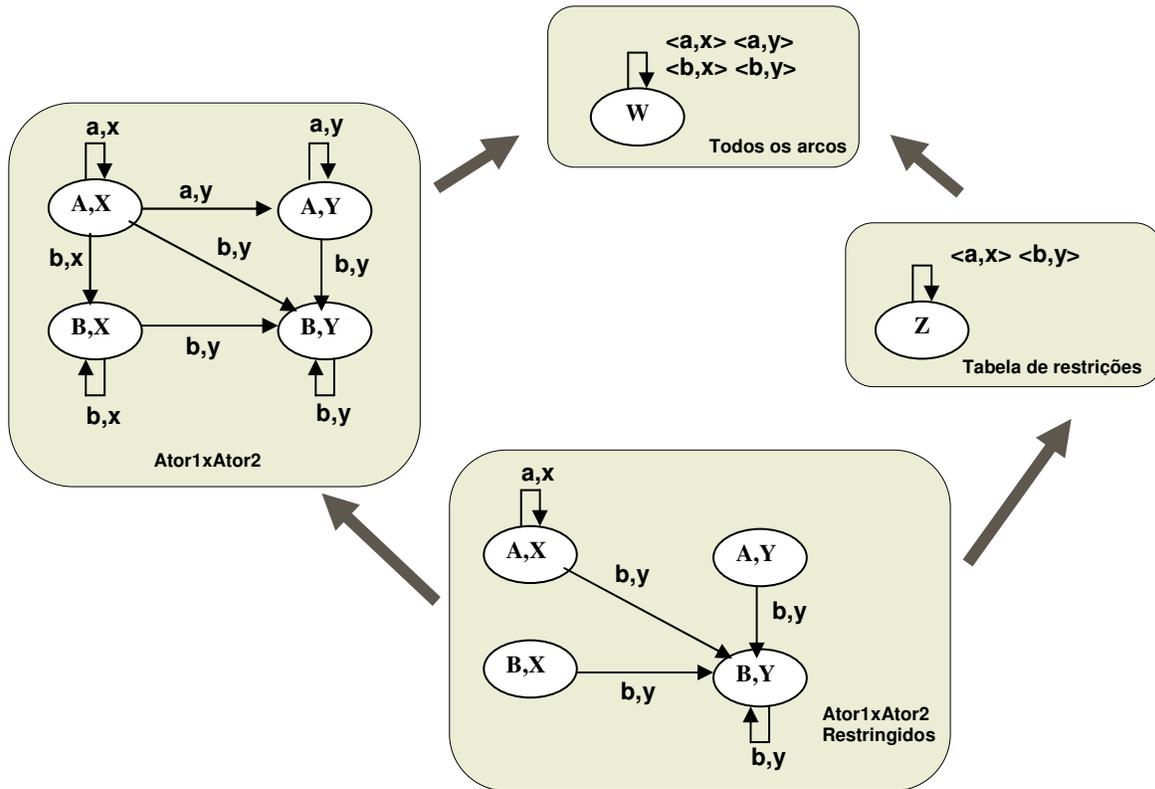


Figura 7.1: Produto Fibrado de grafos como operador do tipo restrição

Como dito anteriormente, uma Soma Amalgamada pode ser interpretada como uma forma de “somar” dois objetos identificando partes em comum. O diagrama que representa a soma amalgamada como uma composição de sistemas identificando partes em comum é apresentado na figura 7.2. Um exemplo de Soma Amalgamada como composição de sistemas identificando partes em comum pode ser encontrado no livro do Menezes [2002, pág. 204].

A soma amalgamada de grafos poderia ser calculada usando as construções de coproduto e de co-equalizador, uma explicação mais didática desse cálculo é encontrada em Menezes [2002, pág. 204], teorema 6.23- Soma Amalgamada em Gr.

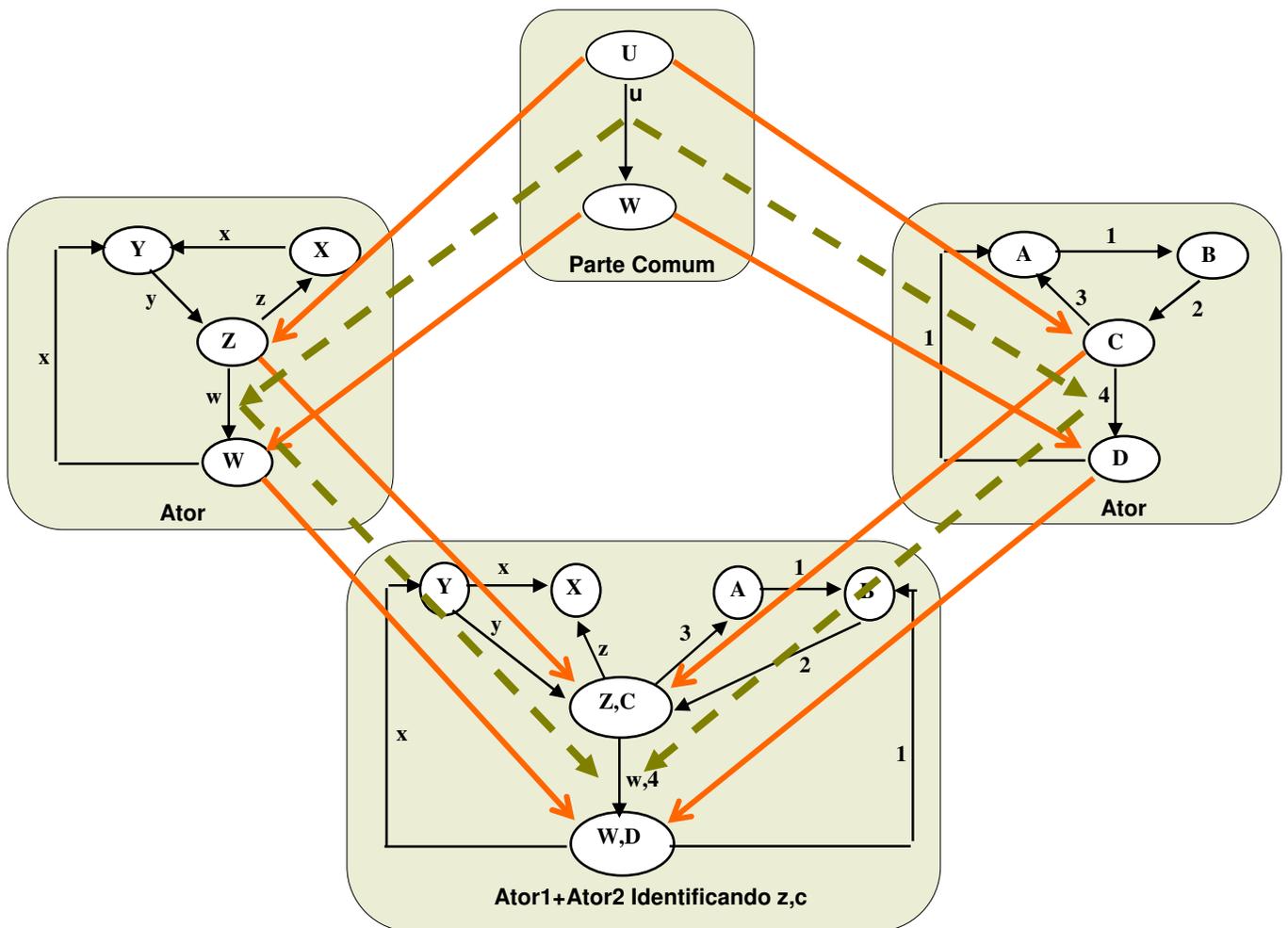


Figura 7.2: Soma Amalgamada de grafos como operador de sistemas

7.3 Produto Fibrado como restrição de sistema

A aplicação do Produto Fibrado como restrição de sistema entre atores de uma animação por computador, baseada em autômato finito, resulta em um novo ator criado a partir dos autômatos dos atores 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.

Para o exemplo que será apresentado a seguir a operação de Produto Fibrado como restrição de sistema será aplicada sobre a operação categorial de produto. A operação categorial de produto entre os dois atores da figura 7.3 é apresentada na figura 7.4. A forma de como se efetua essa operação já foi explicada no capítulo 6.

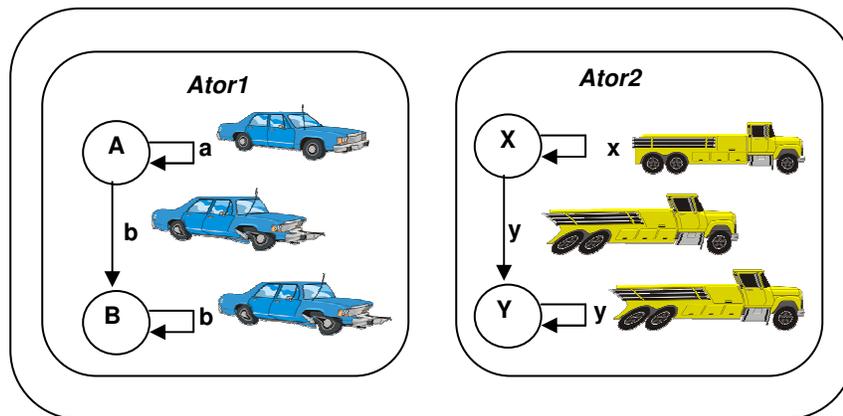


Figura 7.3: Atores de uma animação

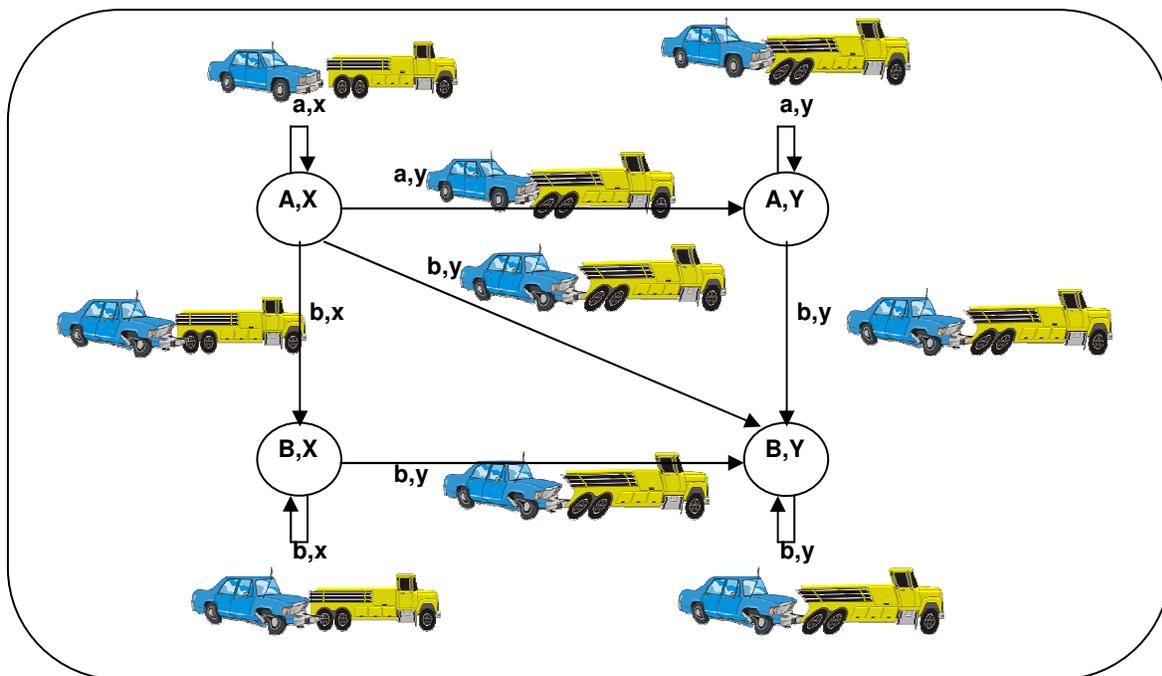


Figura 7.4: Produto entre dois atores de uma animação

Após aplicar a operação categorial de produto entre os atores será definida um restrição. Para ilustrar o referido exemplo tem-se a seguinte restrição:

Restrição: Na animação só deverá aparecer o caminhão em perfeito estado (x) quando o carro também estiver em perfeito estado (a) e o caminhão batido só deverá aparecer quando o carro também estiver batido. Portanto só deverá aparecer o que for $\langle a,x \rangle$ e $\langle b,y \rangle$.

A figura 7.5 mostra o resultado do Produto Fibrado como restrição de sistema a partir da restrição definida.

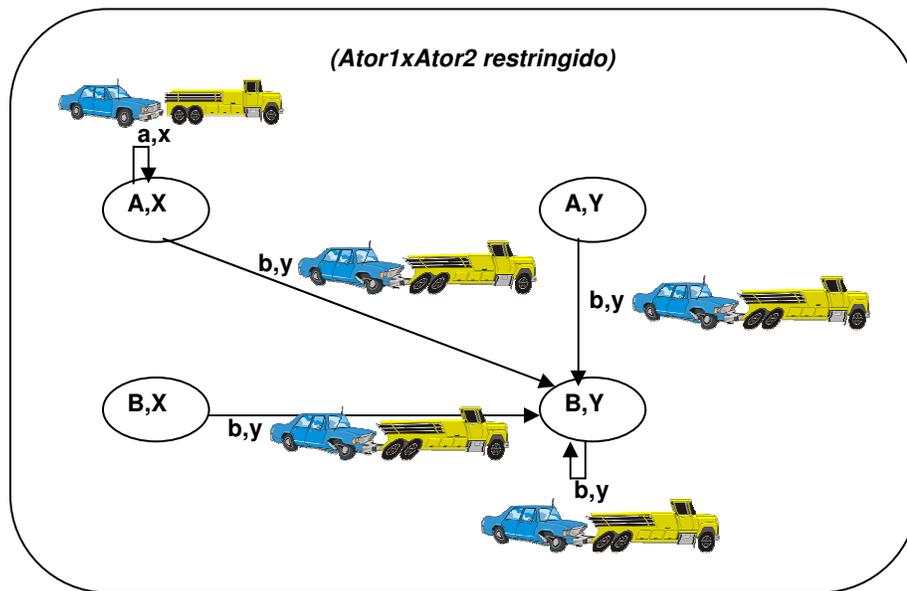


Figura 7.5: Resultado do Produto Fibrado entre dois atores de uma animação

7.3.1 Fita de entrada

Sendo que nessa dissertação a aplicação da operação categorial de Produto Fibrado é feita sobre os autômatos dos atores da animação e não sobre a fita de entrada, torna-se necessário que uma nova fita seja definida.

A fita de entrada é definida como segue:

$$ft = (\langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1), \\ (\langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2) \dots \\ (\langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n)$$

Na qual cada célula possui o os elementos $(\langle a \text{ ator}^1, a \text{ ator}^2 \dots a \text{ ator}^n \rangle, \langle j \text{ ator}^1, j \text{ ator}^2 \dots j \text{ ator}^n \rangle, t)$ onde:

a → um símbolo do alfabeto de entrada

j → cadeia de funções de transformação da saída

t → tempo de transição entre os frames da animação

Para ilustrar o exemplo acima tem-se a fita definida como:

$$ft = (\langle a,x \rangle, \langle \text{Trans}(2,50), \text{Trans}(142,70) \rangle, 200), \\ (\langle a,y \rangle, \langle \text{Trans}(32,0), \text{Trans}(142,70) \rangle, 200), \\ (\langle b,y \rangle, \langle \text{Trans}(32,0), \text{Trans}(142,70) \rangle, 200), \\ (\langle b,y \rangle, \langle \text{Trans}(32,0), \text{Trans}(142,70) \rangle, 200)$$

A Tabela de restrição seria a seguinte:

$\langle a,x \rangle$ e $\langle b,y \rangle$

A animação obtida é mostrada na figura 7.6.

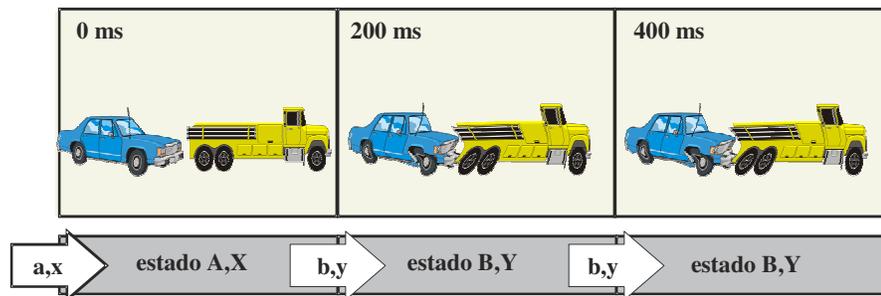


Figura 7.6: Animação do Produto Fibrado como restrição de sistema

Pode-se observar que quando a célula de entrada a que corresponde ao símbolo de entrada não é igual a uma das restrições definidas na tabela de restrições a imagem de saída não é apresentada na área de animação e a próxima célula é lida ignorando-se o tempo de espera da célula anterior. No caso acima o $\langle a,y \rangle$ não é exibido e o seu tempo de espera é ignorado.

Posteriormente será apresentada uma sugestão de implementação do Produto Fibrado como restrição de sistemas.

7.3.2 Comparação com o AGA

No modelo AGA o mesmo resultado do Produto Fibrado como restrição de sistemas pode ser obtido somente se a fita de entrada de cada um dos atores for alterada. Portanto, se tiver 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, nem as fitas e nem os atores originais precisam ser alterados, a animação é obtida utilizando-se a fita definida 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.

7.4 Produto Fibrado como sincronizador de sistemas

O Produto Fibrado também pode ser interpretado como um sincronizador de sistemas. Como exemplo será utilizado o autômato ilustrado na figura 6.15, que representa um ator baseado em grafos reflexivos.

Será desejado que seja sincronizado o homem chutando a bola (b) com o goleiro caído e a bola entrando (y), isso significa dizer que eles devem ocorrer simultaneamente, as outras figuras, a,x , podem ocorrer livremente.

A Tabela de sincronização seria a seguinte:

$\langle b,y \rangle$

$\langle a,tx \rangle, \langle a,ty \rangle, \langle ta,x \rangle, \langle a,x \rangle, \langle a,ty \rangle, \langle tb,x \rangle$

A figura 7.7 ilustra como ficaria o produto fibrado com as sincronizações desejadas. Para simplificar as saídas da figura 7.7 não serão desenhadas, mas elas tem relação com a figura 6.14.

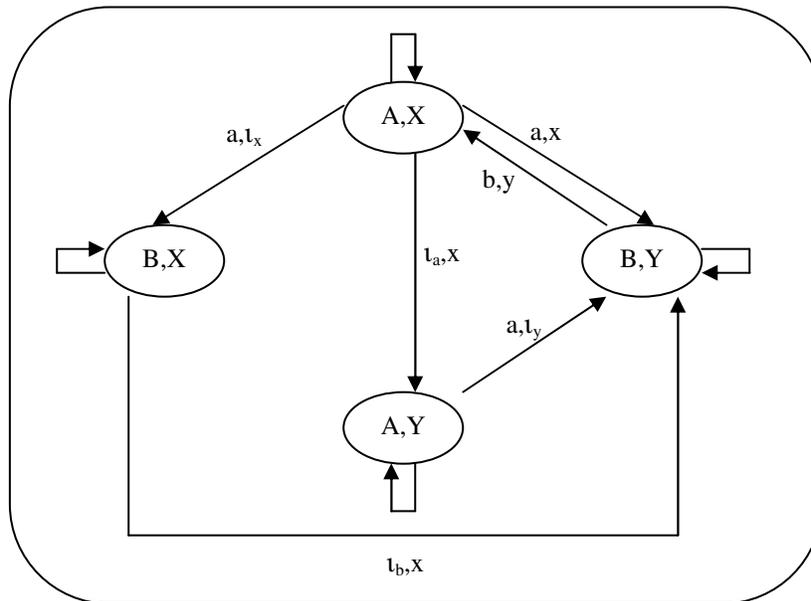


Figura 7.7: Produto Fibrado como sincronizador de sistema

7.4.1 Fita de entrada

Sendo que nessa dissertação a aplicação da operação categorial de Produto Fibrado é feita sobre os autômatos dos atores da animação e não sobre a fita de entrada, torna-se necessário que uma nova fita seja definida.

A fita de entrada é definida como segue:

$$ft = (\langle a_1 ator^1, a_1 ator^2 \dots a_1 ator^n \rangle, \langle j_1 ator^1, j_1 ator^2 \dots j_1 ator^n \rangle, t_1), \\ (\langle a_2 ator^1, a_2 ator^2 \dots a_2 ator^n \rangle, \langle j_2 ator^1, j_2 ator^2 \dots j_2 ator^n \rangle, t_2) \dots \\ (\langle a_n ator^1, a_n ator^2 \dots a_n ator^n \rangle, \langle j_n ator^1, j_n ator^2 \dots j_n ator^n \rangle, t_n)$$

Na qual cada célula possui o os elementos ($\langle a ator^1, a ator^2 \dots a ator^n \rangle, \langle j ator^1, j ator^2 \dots j ator^n \rangle, t$) onde:

$a \rightarrow$ um símbolo do alfabeto de entrada

$j \rightarrow$ cadeia de funções de transformação da saída

$t \rightarrow$ tempo de transição entre os frames da animação

Para ilustrar o exemplo acima tem-se a fita definida como:

$$ft = (\langle a, \varepsilon \rangle, \langle Trans(2,50), \varepsilon \rangle, 200), \\ (\langle b, x \rangle, \langle Trans(32,0), Trans(142,70) \rangle, 200), \\ (\langle \varepsilon, x \rangle, \langle \varepsilon, Trans(142,70) \rangle, 200), \\ (\langle b, y \rangle, \langle Trans(32,0), Trans(142,70) \rangle, 200)$$

A Tabela de sincronização seria a seguinte:

$\langle b, y \rangle$

$\langle a, t_x \rangle, \langle a, t_y \rangle, \langle t_{a,x} \rangle, \langle a, x \rangle, \langle a, t_y \rangle, \langle t_{b,x} \rangle$

A animação obtida é mostrada na figura 7.8.

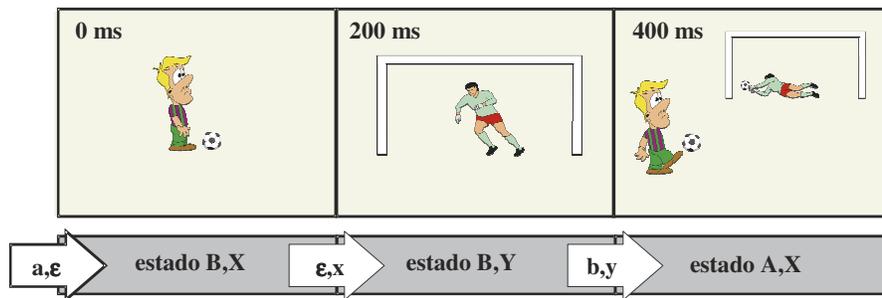


Figura 7.8: Animação do Produto Fibrado como sincronizador de sistema

Pode-se observar que quando a célula de entrada ($\mathbf{b,x}$), que corresponde ao símbolo de entrada, não é igual a uma das sincronizações definidas na tabela de sincronização a imagem de saída não é apresentada na área de animação e a próxima célula é lida ignorando-se o tempo de espera da célula anterior. No caso acima o $\langle \mathbf{b,x} \rangle$ não é exibido e o seu tempo de espera é ignorado.

Posteriormente será apresentada uma sugestão de implementação do Produto Fibrado como sincronização de sistemas.

7.5 Soma Amalgamada sem identificação das saídas

A operação categorial de Soma Amalgamada como composição de um sistema identificando partes em comum será explorada no presente capítulo. A figura 7.9 ilustra os autômatos de dois atores que serão utilizados para exemplificar a Soma Amalgamada.

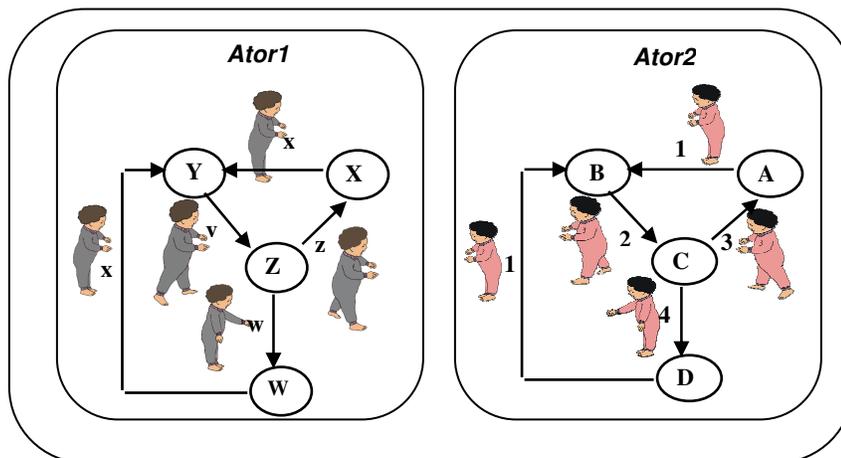


Figura 7.9: Atores de uma animação

Uma das interpretações dada a Soma Amalgamada é a identificação das partes que serão consideradas comuns entre os autômatos envolvidos na operação, identificando ou não a saída.

Para o exemplo a ser apresentado será necessário que se identifique quais partes do autômato serão consideradas comum, porém não será identificada a saída.

Para os dois atores apresentados na figura 7.9 a parte em comum será o w com o 4 ($w,4$). Uma das interpretações dada a operação categorial de Soma Amalgamada é que a parte em comum é somada, ou seja, unida e o resto do autômato de cada ator envolvido na operação é independente. O resultado final dessa interpretação será um novo autômato conforme apresentado na figura 7.10.

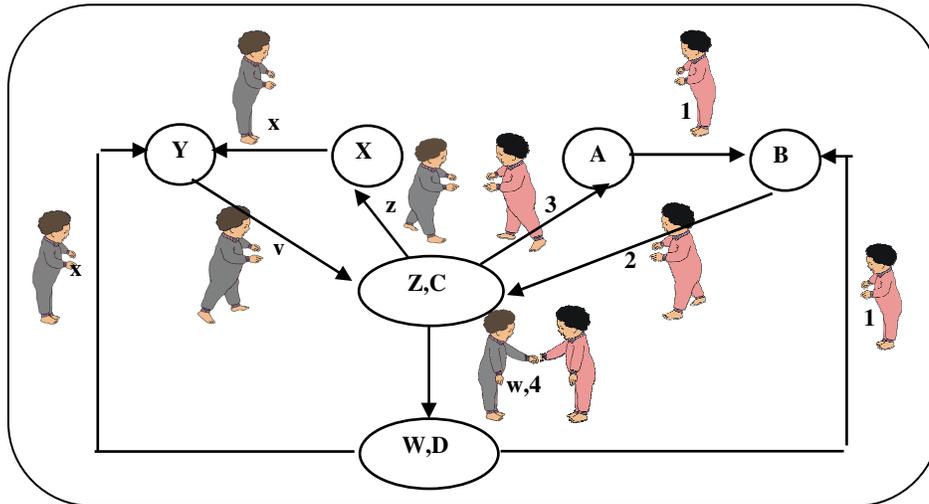


Figura 7.10: Soma Amalgamada entre dois atores de uma animação

7.5.1 Fita de entrada

Na aplicação da operação categorial de Soma Amalgamada sobre dois ou mais atores de uma animação a fita de entrada será única para os atores envolvidos na animação. A fita seria a seguinte:

$$ft = \langle \langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1 \rangle, \\ \langle \langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2 \rangle \dots \\ \langle \langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n \rangle$$

Na qual cada célula possui os elementos $\langle \langle a \text{ ator}^1, a \text{ ator}^2 \dots a \text{ ator}^n \rangle, \langle j \text{ ator}^1, j \text{ ator}^2 \dots j \text{ ator}^n \rangle, t \rangle$ onde:

a → um símbolo do alfabeto de entrada

j → cadeia de funções de transformação da saída

t → tempo de transição entre os frames da animação

O resultado da animação está representado pela figura 7.11. A fita de entrada poderá ser considerada a seguinte:

$$ft = \langle \langle 1, \varepsilon \rangle, \langle \text{Trans}(142,50), \varepsilon \rangle, 200 \rangle, \\ \langle \langle 2, \varepsilon \rangle, \langle \text{Trans}(122,50), \varepsilon \rangle, 200 \rangle, \\ \langle \langle 3, \varepsilon \rangle, \langle \text{Trans}(102,50), \varepsilon \rangle, 200 \rangle, \\ \langle \langle 1, \varepsilon \rangle, \langle \text{Trans}(82,50), \varepsilon \rangle, 200 \rangle, \\ \langle \langle 2, \varepsilon \rangle, \langle \text{Trans}(52,50), \varepsilon \rangle, 200 \rangle, \\ \langle \langle 4, w \rangle, \langle \text{Trans}(20,50), \text{Trans}(32,85) \rangle, 200 \rangle, \\ \langle \langle \varepsilon, x \rangle, \langle \varepsilon, \text{Trans}(42,50) \rangle, 200 \rangle, \\ \langle \langle \varepsilon, y \rangle, \langle \varepsilon, \text{Trans}(62,50) \rangle, 200 \rangle$$

O símbolo ε significa que não se tem elemento na respectiva célula.

É preciso definir o que é considerado igual, para este exemplo o símbolo w com o símbolo 4 é considerado igual, isto significa dizer que as saídas devem ser exibidas na mesma tela de animação e com o mesmo tempo de duração respeitando as transformações de cada uma delas.

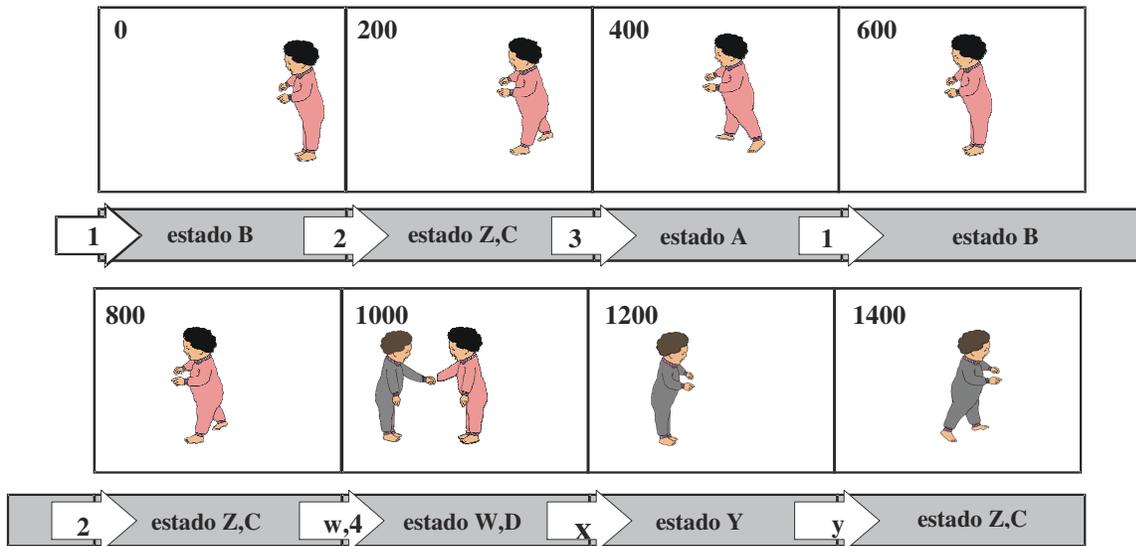


Figura 7.11: Animação da Soma Amalgamada envolvendo dois atores

7.6 Soma Amalgamada com identificação das saídas

Para representar a soma amalgamada com identificação das saídas, será utilizado como exemplo os dois atores apresentados na figura 7.12.

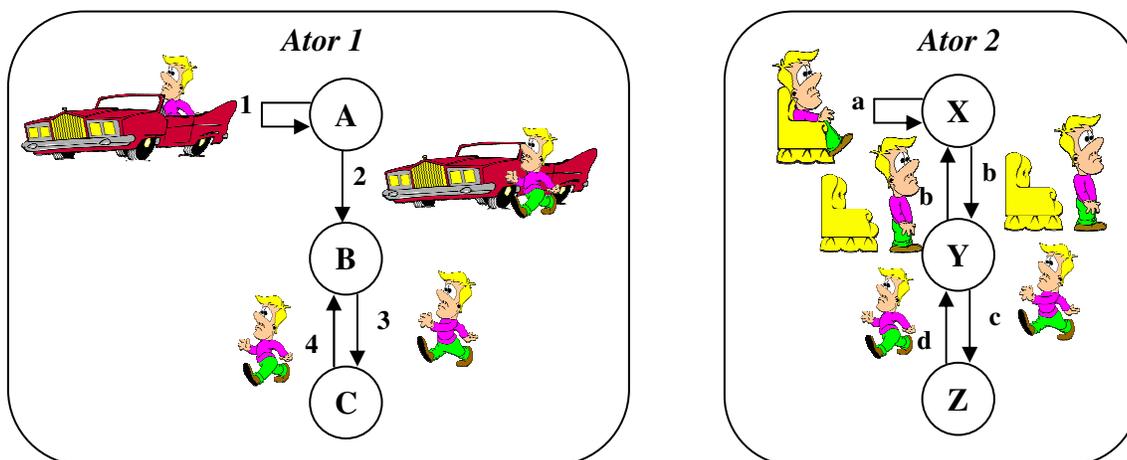


Figura 7.12: Atores de uma animação

Para o exemplo a ser apresentado será necessário que se identifique quais partes do autômato serão consideradas comum identificando a saída. Para os dois atores apresentados na figura 7.12 a parte em comum será o 3 com o c (3,c) e o 4 com o d (4,d). Uma das

interpretações dada a operação categorial de Soma Amalgamada é que a parte em comum com a mesma saída é somada, ou seja, unida apenas uma imagem é exibida na tela de apresentação e o resto do autômato de cada ator envolvido na operação é independente. O resultado final dessa interpretação será um novo autômato conforme apresentado na figura 7.13.

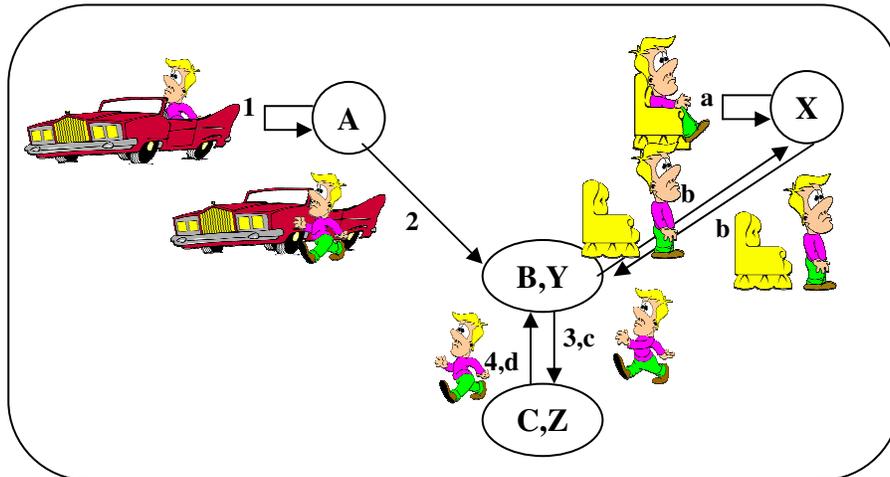


Figura 7.13: Soma Amalgamada identificando as saídas

A definição da fita de entrada para a soma amalgamada com identificação das saídas e a mesma da soma amalgamada sem a identificação das saídas explicada no capítulo 7.5.1.

O resultado da animação sendo está representado pela figura 7.14. A fita de entrada é a seguinte:

$$ft = \langle 1, \varepsilon, \langle \text{Trans}(142, 50), \varepsilon, 200 \rangle, \langle 2, \varepsilon, \langle \text{Trans}(122, 50), \varepsilon, 200 \rangle, \langle 3, \varepsilon, \langle \text{Trans}(102, 50), \varepsilon, 200 \rangle, \langle \varepsilon, d, \langle \varepsilon, \text{Trans}(82, 50) \rangle, 200 \rangle, \langle \varepsilon, b, \langle \varepsilon, \text{Trans}(52, 50) \rangle, 200 \rangle, \langle \varepsilon, a, \langle \varepsilon, \text{Trans}(20, 50) \rangle, 200 \rangle, \langle \varepsilon, b, \langle \varepsilon, \text{Trans}(42, 50) \rangle, 200 \rangle, \langle 3, \varepsilon, \langle \text{Trans}(62, 50), \varepsilon, 200 \rangle \rangle \rangle \rangle$$

O símbolo ε significa que não se tem elemento na respectiva célula da fita de entrada.

Sendo que as partes consideradas iguais são: $4=d$ e $3=c$, isto quer dizer que as imagens de saída são as mesmas, por isso tanto faz fazer referência ao símbolo de entrada 4 ou ao símbolo de entrada d , pois os dois possuem imagens de saída iguais.

Portanto na fita de entrada pode-se tanto entrar com o 4 ou com d e com 3 ou c , a imagem exibida na tela de apresentação será a mesma. A figura 7.14 ilustra como ficará a apresentação da soma amalgamada representada na figura 7.13, com a leitura da fita de entrada descrita.

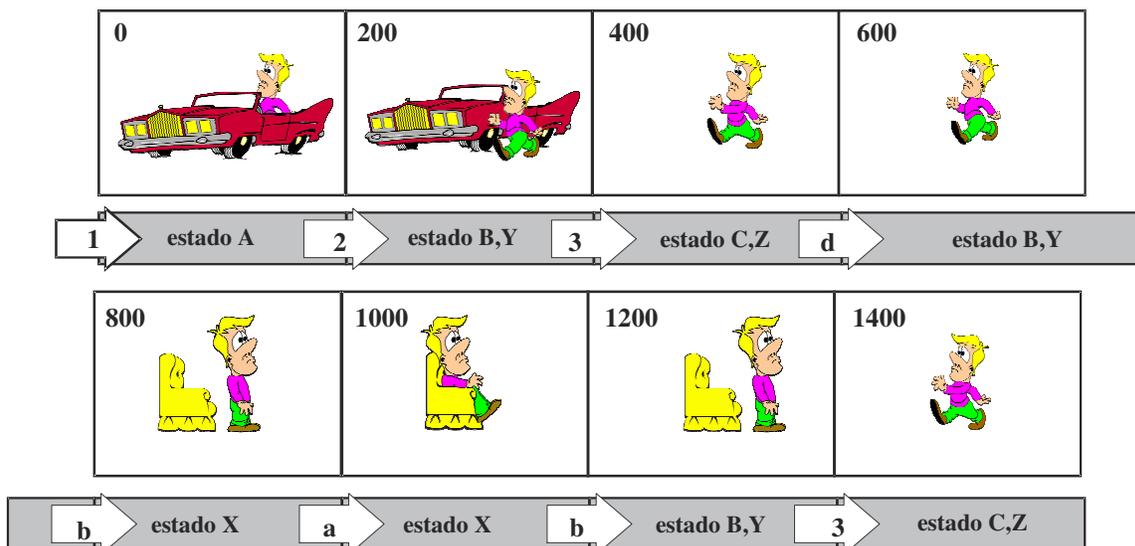


Figura 7.14: Animação da Soma Amalgamada envolvendo dois atores

7.7 Composição de Operações

É possível que se aplique operações categoriais sobre operações categoriais, quantas vezes se ache necessário para que se obtenha o resultado desejado. Como exemplo será apresentado pelas figuras 7.16 e 7.17 as operações categoriais de Soma amalgamada sem identificação das saída e Produto, entre os atores ilustrados na figura 7.15.

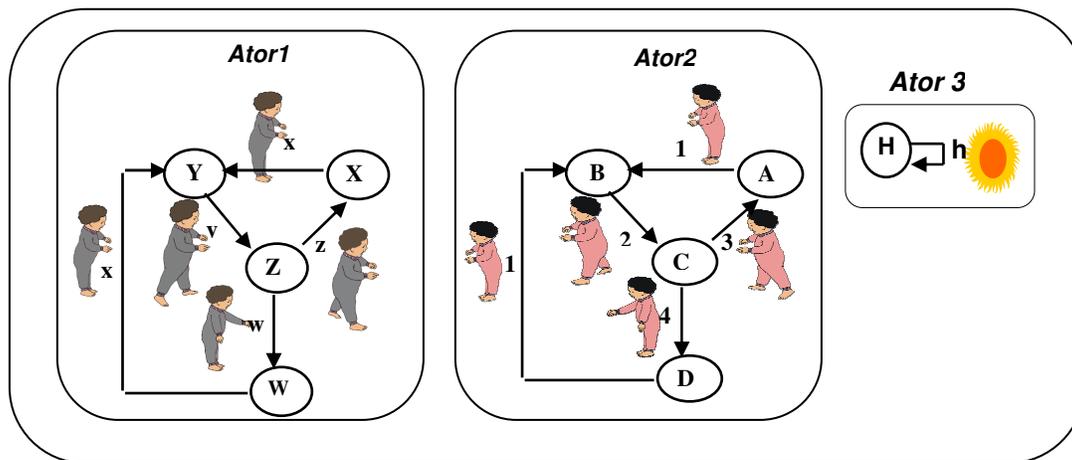


Figura 7.15: Atores de uma animação

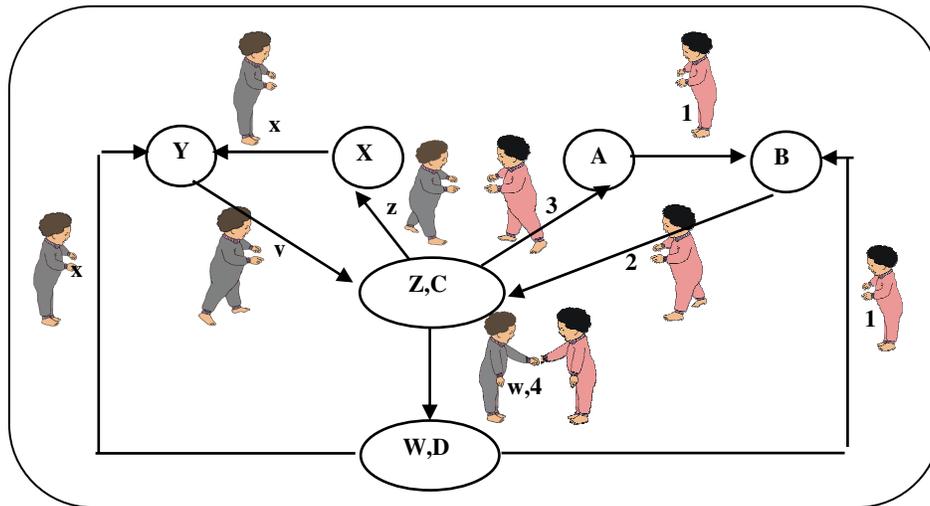


Figura 7.16: Soma Amalgamada, sem identificação das saídas, entre ator1 e ator2

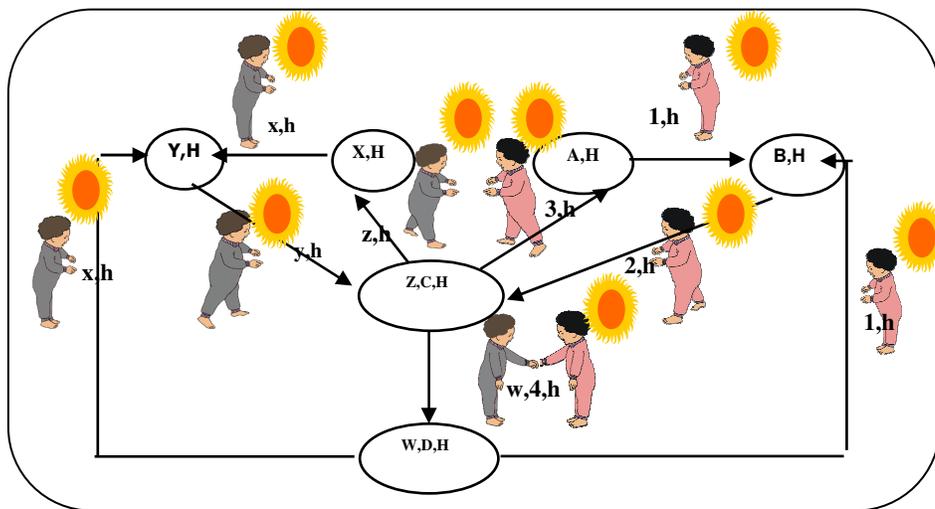


Figura 7.17: (Soma Amalgamada, sem identificação das saídas, entre ator1 e ator2)
x Ator3

7.8 Considerações Finais e Trabalhos Futuros

Neste capítulo se estudou as operações categoriais de Produto Fibrado e Soma Amalgamada aplicadas em grafos. As animações computacionais serviram para verificar o resultado prático da aplicação dessas operações.

Pôde-se observar que tanto no Produto Fibrado como na Soma Amalgamada o objeto resultante é sempre um novo ator gerado a partir dos atores originais, portanto se o autômato original for modificado o resultado das operações será alterado automaticamente. Com a definição das fitas de entrada para cada uma das operações é possível que se obtenha uma nova animação a partir dos atores originais.

No estudo do Produto Fibrado foram verificados os resultados sob duas interpretações Produto Fibrado como restrição de um sistema e Produto Fibrado como sincronizador de um sistema. No estudo da Soma Amalgamada também foram estudadas duas interpretações Soma Amalgamada com identificação das saídas e Soma Amalgamada sem identificação das saídas.

O capítulo mostra ainda que é possível aplicar operações categoriais sobre operações categoriais quantas vezes se ache necessário.

O capítulo não trata da aplicação das operações categoriais sobre as fitas de entrada das animações, este é um tema que poderá ser explorado em um trabalho futuro.

O objetivo do presente capítulo não é provar que a operação categorial de Produto Fibrado e Soma Amalgamada é a melhor ou a pior a maneira encontrada para se obter combinações entre os atores de uma animação. O que se investigou aqui é a fundamentação matemática para que se possa fazer combinações entre os atores de uma animação.

8 SUGESTÃO DE IMPLEMENTAÇÃO

8.1 Introdução

O objetivo deste trabalho não é implementar as operações categoriais aqui estudadas. Será apresentada uma sugestão de como essas operações poderão ser implementadas tomando como base a implementação do modelo AGA, fazendo-se uma extensão da implementação do protótipo do modelo AGA.

Para cada uma das operações categoriais estudadas será apresentado uma sugestão de implementação, porém, pode-se encontrar outras formas de implementação que não sejam as que estão aqui definidas e que também poderão ser utilizadas.

É importante ressaltar que no estudo apresentado as operações categoriais estudadas foram aplicadas apenas sobre os atores da animação e não sobre as fitas de entrada, por isto a estrutura original das fitas de entradas, definidas por Accorsi, serão mantidas.

O que se sugere é um mapeamento das fitas definidas em cada uma das operações categoriais para a fita definida por Accorsi.

8.2 Visão Geral

Para cada ator de uma animação existe um alfabeto de entrada. A fita de entrada deverá estar de acordo com este alfabeto e ter consistência, para que se possa obter uma animação. A figura 8.1 ilustra como são estas fitas e como é o alfabeto de entrada.

Nesta dissertação, para cada uma das operações categoriais estudadas foi definida uma nova fita de entrada, a fita é única, independente do número de atores envolvidos na operação. No modelo AGA, as fitas são individuais para cada um dos atores. Portanto o objetivo é que se mapeie a fita da respectiva operação nas fitas individuais de cada um dos atores envolvidos. A figura 8.2 ilustra o mapeamento das fitas.

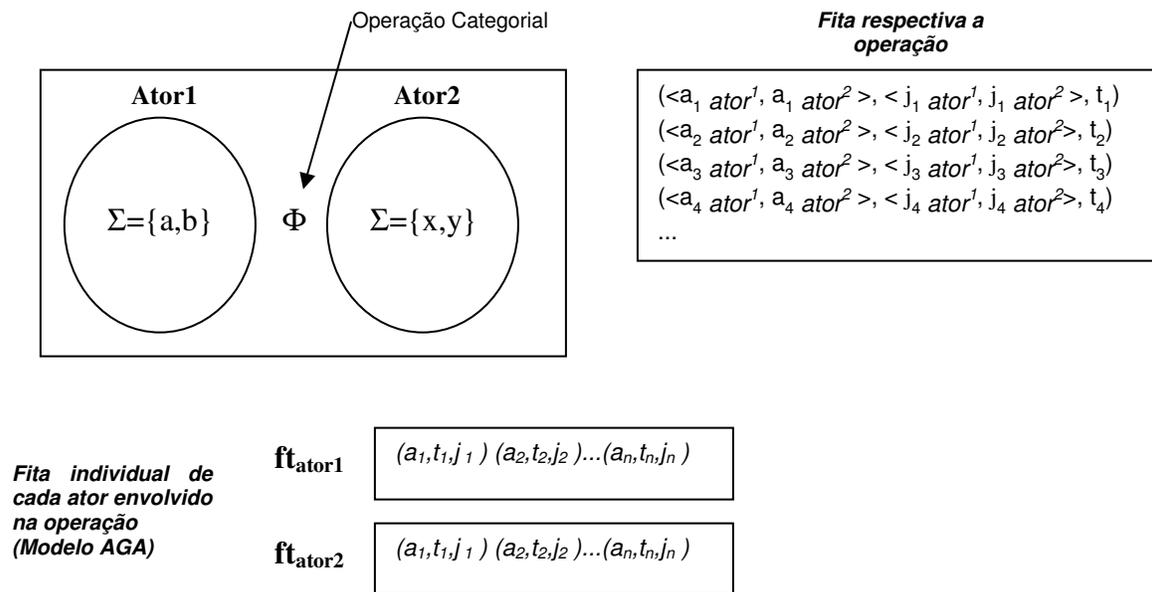


Figura 8.1: Ilustração da implementação

A fita de entrada única, da operação categorial, respeita os elementos e o funcionamento pré-definidos no modelo AGA [ACC 2002].

Os elementos existentes, tanto na fita única, como na fita individual de cada um dos atores, são os mesmos:

a → um símbolo do alfabeto de entrada

j → cadeia de funções de transformação da saída

t → tempo de transição

O elemento tempo (**t**), tem um enfoque diferente para as fitas. Na fita individual, definida no modelo AGA, o elemento tempo é específico de cada um dos atores da animação, isto significa que em um mesmo quadro em que aparece mais de um ator, um deles poderá ser exibido por 200 milésimos de segundo enquanto o outro for exibido por 100 milésimos de segundo. Já na fita única, definida para a operação categorial, o elemento tempo é único para todos os atores envolvidos na operação, ou seja, se for definido 200 milésimos de segundo, todos os atores naquele quadro são exibidos pelo mesmo tempo. Quando esse elemento for mapeado, para as fitas individuais, ele deverá ser o mesmo para fita de cada um dos atores envolvidos na operação. Os outros elementos serão específicos de cada um dos atores, conforme ilustra a figura 8.2.

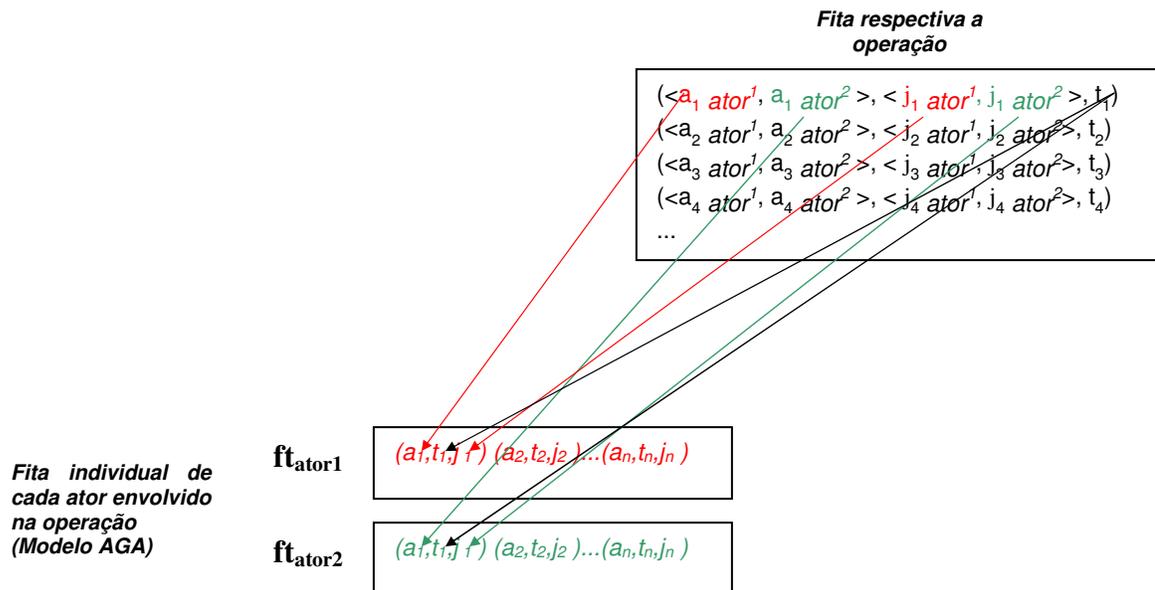


Figura 8.2: Mapeamento das fitas de entrada

Depois de efetuado o mapeamento das fitas a implementação da animação é feita conforme definido no modelo AGA [ACC 2002] e no modelo AGA Player [MAH 2002].

8.3 Produto

A fita definida para a operação categorial de Produto é a seguinte:

$$ft = (\langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1), \\ (\langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2) \dots \\ (\langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n)$$

Para os atores apresentados na figura 6.3, aplicando-se a operação de Produto tem-se como exemplo a seguinte fita:

$$ft = (\langle 1, a \rangle, \langle \xi, \epsilon, 100 \rangle) \\ (\langle 3, b \rangle, \langle \xi, \epsilon, 100 \rangle) \\ (\langle 2, c \rangle, \langle \xi, \epsilon, 100 \rangle)$$

A fita mapeada para cada um dos atores seria:

$$ft_{\text{ator1}} = (1, \xi, 100) (3, \xi, 100) (2, \xi, 100) \\ ft_{\text{ator2}} = (a, \xi, 100) (b, \xi, 100) (c, \xi, 100)$$

Conforme definido pelo modelo AGA e no AGAPlayer, a leitura da fita é feita simultaneamente e os atores são apresentados na tela de animação ao mesmo tempo respeitando as camadas de sobreposição. A animação obtida pela leitura da fita está representada na figura 8.3

Fita de Entrada		
a	j	t
1	ϵ	100
a	ϵ	
3	ϵ	100
b	ϵ	
2	ϵ	100
c	ϵ	

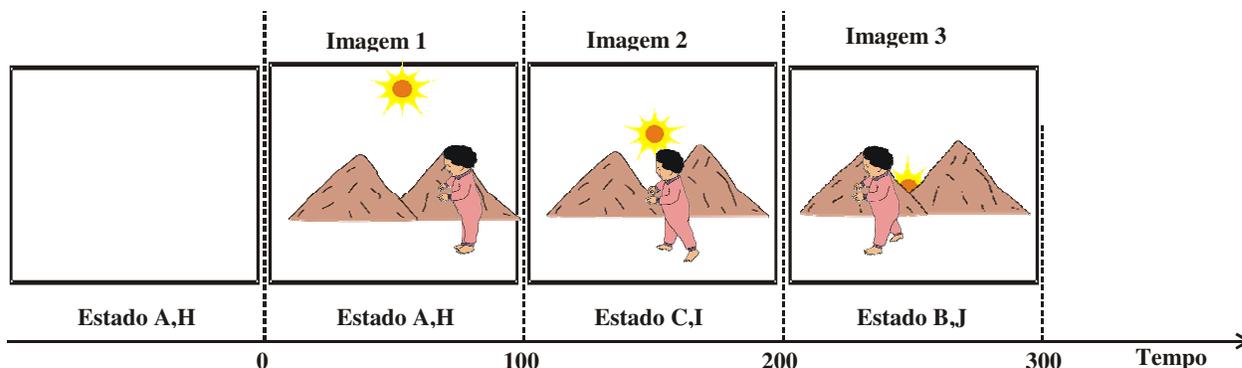


Figura 8.3: Animação obtida pela leitura da fita

8.4 Coproduto

A fita de entrada definida para a operação categorial de Coproduto é a seguinte:

$$ft = (\langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1), \\ (\langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2) \dots \\ (\langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n)$$

Para os atores apresentados na figura 6.3, aplicando-se a operação de Coproduto tem-se como exemplo a seguinte fita:

$$ft = (\langle 1, a \rangle, \langle \epsilon, \epsilon \rangle, 100) \\ (\langle 3, b \rangle, \langle \epsilon, \epsilon \rangle, 100) \\ (\langle 2, c \rangle, \langle \epsilon, \epsilon \rangle, 100)$$

A fita mapeada para cada um dos atores seria:

$$ft_{\text{ator1}} = (1, \epsilon, 100) (3, \epsilon, 100) (2, \epsilon, 100) \\ ft_{\text{ator2}} = (a, \epsilon, 100) (b, \epsilon, 100) (c, \epsilon, 100)$$

Como está se tratando da operação categorial de Coproduto, em que “o Coproduto caracteriza um não determinismo ou escolha, no sentido em que cada um dos grafos componentes do objeto resultante do coproduto representa uma alternativa [MEN 2002]”, então um dos atores poderá ser escolhido para ser exibido na tela ou um é exibido e depois o outro. Neste exemplo somente o ator1 foi exibido, conforme ilustra a figura 8.4.

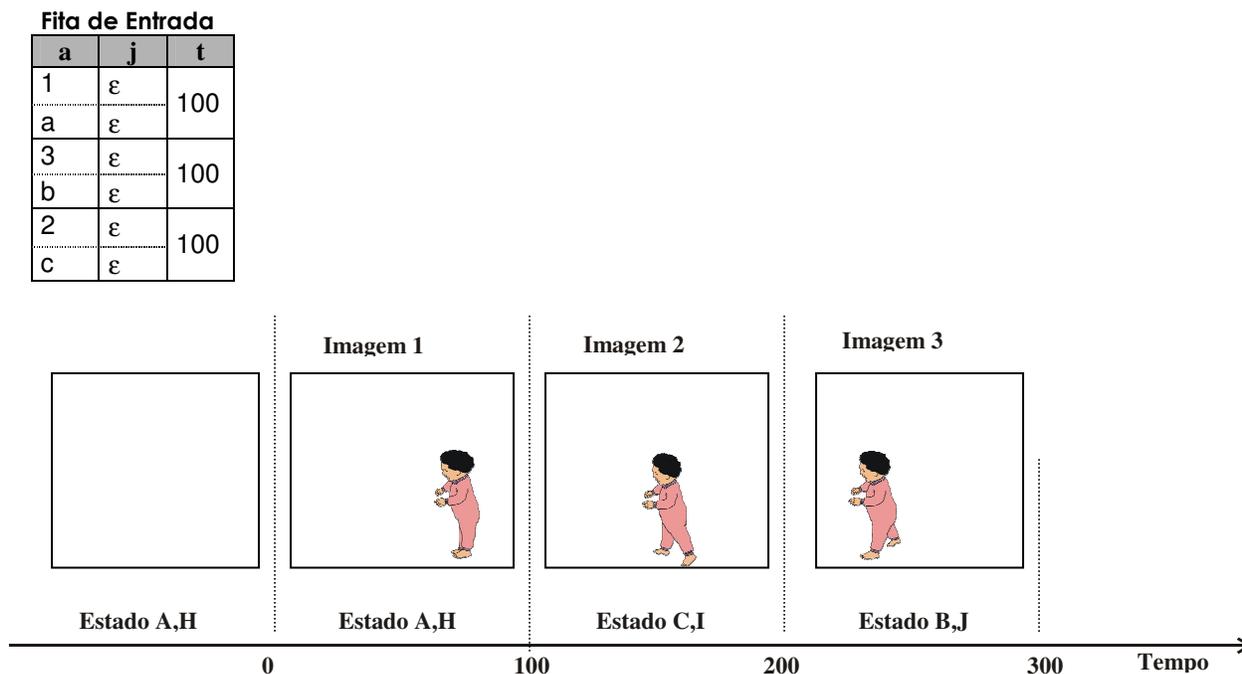


Figura 8.4: Animação obtida pela leitura da fita

A escolha poderá ser definida na implementação de forma aleatória, identificando um não determinismo interno. Isto significa escolher uma das fitas ou escolher primeiro uma fita para ser processada e depois a outra fita, então a ordem das fitas a serem processadas seria escolhida aleatoriamente.

A escolha também poderá ser feita pelo usuário, identificando um não determinismo externo, pois dependendo do usuário a escolha é diferente.

8.5 Produto Fibrado como restrição de sistema

A fita de entrada para a operação categorial de Produto Fibrado como restrição de sistema é definida como segue:

$$ft = (\langle a_1 ator^1, a_1 ator^2 \dots a_1 ator^n \rangle, \langle j_1 ator^1, j_1 ator^2 \dots j_1 ator^n \rangle, t_1), \\ (\langle a_2 ator^1, a_2 ator^2 \dots a_2 ator^n \rangle, \langle j_2 ator^1, j_2 ator^2 \dots j_2 ator^n \rangle, t_2) \dots \\ (\langle a_n ator^1, a_n ator^2 \dots a_n ator^n \rangle, \langle j_n ator^1, j_n ator^2 \dots j_n ator^n \rangle, t_n)$$

Para os atores apresentados na figura 7.3, aplicando-se a operação de Produto Fibrado como restrição de sistema tem-se como exemplo a seguinte fita:

$$ft = (\langle a,x \rangle, \langle Trans(2,50), Trans(142,70) \rangle, 200) \\ (\langle a,y \rangle, \langle Trans(32,0), Trans(142,70) \rangle, 200) \\ (\langle b,y \rangle, \langle Trans(32,0), Trans(142,70) \rangle, 200) \\ (\langle b,y \rangle, \langle Trans(32,0), Trans(142,70) \rangle, 200)$$

A Tabela de restrição seria a seguinte:

$\langle a,x \rangle$ e $\langle b,y \rangle$

A fita mapeada para cada um dos atores seria a seguinte:

$$f_{ator1} = (a, Trans(2,50), 200) (a, Trans(32,0), 200) (b, Trans(32,0), 200)$$

$(b, \text{Trans}(32,0), 200)$

$f_{\text{ator2}} = (x, \text{Trans}(142,70), 200) (y, \text{Trans}(142,70), 200) (y, \text{Trans}(142,70), 200)$

$(y, \text{Trans}(142,70), 200)$

Obedecendo a *Tabela de restrição* em que o primeiro elemento de cada uma das células fazem referência ao ator1 (a,b) e os segundos elementos de cada célula fazem referência ao ator2 (x,y), a figura 8.5 ilustra a animação obtida com a leitura da fita.

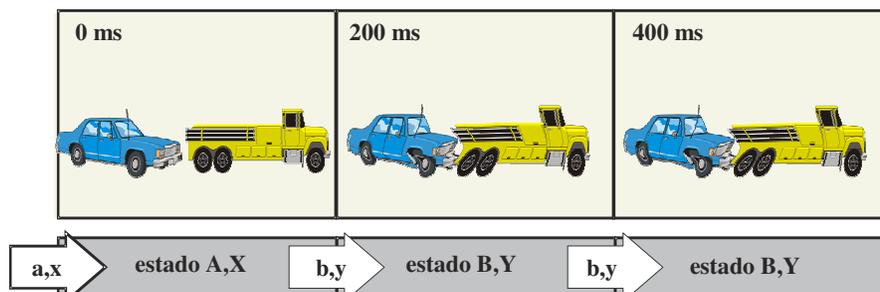


Figura 8.5: Animação obtida pela leitura da fita

Pode-se observar que quando a célula de entrada a que corresponde ao símbolo de entrada não é igual a uma das restrições definidas na tabela de restrições a imagem de saída não é apresentada na área de animação e a próxima célula é lida ignorando-se o tempo de espera da célula anterior. No caso acima o $\langle a,y \rangle$ não é exibido e o seu tempo de espera é ignorado.

8.6 Produto Fibrado como sincronizador de sistema

O Produto Fibrado também pode ser interpretado como um sincronizador de sistema. A fita de entrada para a operação categorial de Produto Fibrado como sincronizador de sistema é definida como segue:

$$ft = (\langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1), \\ (\langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2) \dots \\ (\langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n)$$

Para os atores apresentados na figura 6.14, aplicando-se a operação de Produto Fibrado como sincronizador de sistema tem-se como exemplo a seguinte fita:

$$ft = (\langle a, \varepsilon \rangle, \langle \text{Trans}(2,50), \varepsilon \rangle, 200) \\ (\langle b, x \rangle, \langle \text{Trans}(32,0), \text{Trans}(142,70) \rangle, 200) \\ (\langle \varepsilon, x \rangle, \langle \varepsilon, \text{Trans}(142,70) \rangle, 200) \\ (\langle b, y \rangle, \langle \text{Trans}(32,0), \text{Trans}(142,70) \rangle, 200)$$

A *Tabela de sincronização* seria a seguinte:

$\langle b, y \rangle$

$\langle a, ix \rangle, \langle a, iy \rangle \langle ia, x \rangle \langle a, x \rangle, \langle a, iy \rangle \langle ib, x \rangle$

A fita mapeada para cada um dos atores seria a seguinte:

$f_{\text{ator1}} = (a, \text{Trans}(2,50), 200) (b, \text{Trans}(32,0), 200) (\varepsilon, \varepsilon, 200) \\ (b, \text{Trans}(32,0), 200)$

$f_{\text{ator2}} = (\varepsilon, \varepsilon, 200) (x, \text{Trans}(142,70), 200) (x, \text{Trans}(142,70), 200) \\ (y, \text{Trans}(142,70), 200)$

Obedecendo a *Tabela de sincronização* em que o primeiro elemento de cada uma das células fazem referência ao **Ator1** e o segundo elemento de cada célula fazem referência ao **Ator2**, a figura 8.6 ilustra a animação obtida com a leitura da fita.

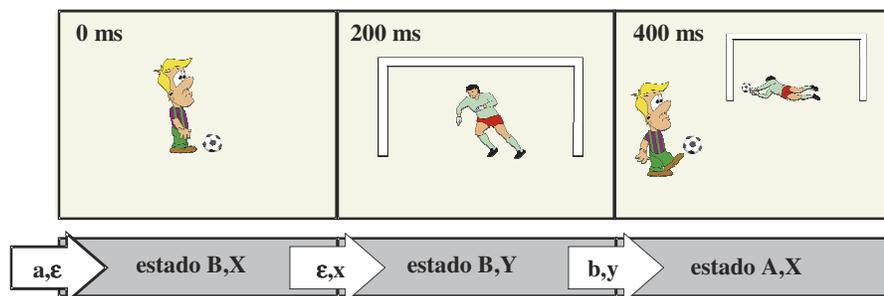


Figura 8.6: Animação obtida pela leitura da fita

8.7 Soma Amalgamada sem identificação das saídas

A fita de entrada para a operação categorial de Soma Amalgamada sem identificação das saídas é definida como segue:

$$ft = (\langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1), \\ (\langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2) \dots \\ (\langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n)$$

Para os atores apresentados na figura 7.9, aplicando-se a operação de Soma Amalgamada tem-se como exemplo a seguinte fita:

$$ft = (\langle 1, \phi \rangle, \langle \text{Trans}(142,50), \phi \rangle, 200) \\ (\langle 2, \phi \rangle, \langle \text{Trans}(122,50), \phi \rangle, 200) \\ (\langle 3, \phi \rangle, \langle \text{Trans}(102,50), \phi \rangle, 200) \\ (\langle 1, \phi \rangle, \langle \text{Trans}(82,50), \phi \rangle, 200) \\ (\langle 2, \phi \rangle, \langle \text{Trans}(52,50), \phi \rangle, 200) \\ (\langle 4, w \rangle, \langle \text{Trans}(20,50), \text{Trans}(32,85) \rangle, 200) \\ (\langle \phi, x \rangle, \langle \phi, \text{Trans}(42,50) \rangle, 200) \\ (\langle \phi, y \rangle, \langle \phi, \text{Trans}(62,50) \rangle, 200)$$

O símbolo ϕ significa que não se tem elemento na respectiva célula da fita de entrada.

A fita mapeada para cada um dos atores seria a seguinte:

$$ft_{\text{ator1}} = (1, \text{Trans}(142,50), 200) (2, \text{Trans}(122,50), 200) (3, \text{Trans}(102,50), 200) \\ (1, \text{Trans}(82,50), 200) (2, \text{Trans}(52,50), 200) (4, \text{Trans}(20,50), 200) \\ (\phi, \phi, 200) (\phi, \phi, 200)$$

$$ft_{\text{ator2}} = (\phi, \phi, 200) (w, \text{Trans}(32,85), 200) \\ (x, \text{Trans}(42,50), 200) (y, \text{Trans}(62,50), 200)$$

Símbolos iguais: $w=4$.

Os símbolos considerados iguais são o **w** e o **4**, isto significa dizer que as saídas devem ser exibidas na mesma tela de animação e com o mesmo tempo de duração respeitando as transformações de cada uma delas, conforme ilustra a figura 8.7.

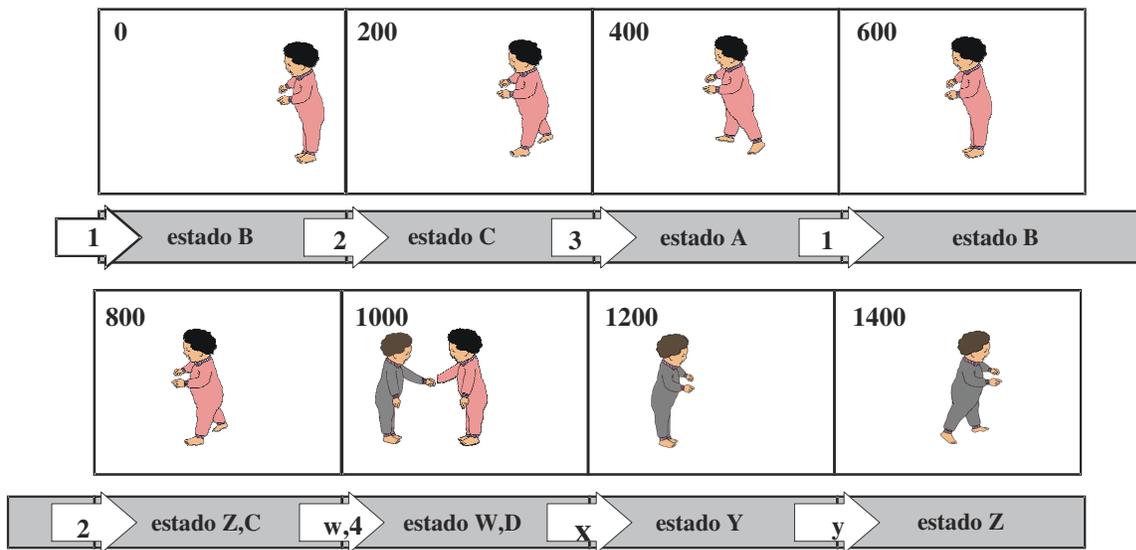


Figura 8.7: Animação obtida pela leitura da fita

8.8 Soma Amalgamada com identificação das saídas

A fita de entrada para a operação categorial de Soma Amalgamada com identificação das saídas é definida como segue:

$$ft = \langle \langle a_1 \text{ ator}^1, a_1 \text{ ator}^2 \dots a_1 \text{ ator}^n \rangle, \langle j_1 \text{ ator}^1, j_1 \text{ ator}^2 \dots j_1 \text{ ator}^n \rangle, t_1 \rangle, \\ \langle \langle a_2 \text{ ator}^1, a_2 \text{ ator}^2 \dots a_2 \text{ ator}^n \rangle, \langle j_2 \text{ ator}^1, j_2 \text{ ator}^2 \dots j_2 \text{ ator}^n \rangle, t_2 \rangle \dots \\ \langle \langle a_n \text{ ator}^1, a_n \text{ ator}^2 \dots a_n \text{ ator}^n \rangle, \langle j_n \text{ ator}^1, j_n \text{ ator}^2 \dots j_n \text{ ator}^n \rangle, t_n \rangle$$

Para os atores apresentados na figura 7.12, aplicando-se a operação de Soma Amalgamada tem-se como exemplo a seguinte fita:

$$ft = \langle \langle 1, \phi \rangle, \langle \text{Trans}(142,50), \phi \rangle, 200 \rangle \\ \langle \langle 2, \phi \rangle, \langle \text{Trans}(122,50), \phi \rangle, 200 \rangle \\ \langle \langle 3, \phi \rangle, \langle \text{Trans}(102,50), \phi \rangle, 200 \rangle \\ \langle \langle \phi, d \rangle, \langle \phi, \text{Trans}(82,50) \rangle, 200 \rangle \\ \langle \langle \phi, b \rangle, \langle \phi, \text{Trans}(52,50) \rangle, 200 \rangle \\ \langle \langle \phi, a \rangle, \langle \phi, \text{Trans}(20,50) \rangle, 200 \rangle \\ \langle \langle \phi, b \rangle, \langle \phi, \text{Trans}(42,50) \rangle, 200 \rangle \\ \langle \langle 3, \phi \rangle, \langle \text{Trans}(62,50), \phi \rangle, 200 \rangle$$

Saídas iguais: $4=d$ e $3=c$.

Portanto na fita de entrada pode-se tanto entrar com o 4 ou com d e com 3 ou c pois as imagens de saída são iguais.

O símbolo ϕ significa que não se tem elemento na respectiva célula da fita de entrada.

A fita mapeada para cada um dos atores seria a seguinte:

$$ft_{\text{ator1}} = \langle \langle 1, \text{Trans}(142,50), 200 \rangle \langle 2, \text{Trans}(122,50), 200 \rangle \langle 3, \text{Trans}(102,50), 200 \rangle \\ \langle \phi, \phi, 200 \rangle \langle 3, \text{Trans}(62,50), 200 \rangle \rangle$$

$$ft_{\text{ator2}} = \langle \langle \phi, \phi, 200 \rangle \langle \phi, \phi, 200 \rangle \langle \phi, \phi, 200 \rangle \langle d, \text{Trnas}(82,50), 200 \rangle \langle b, \text{Trans}(52,50), 200 \rangle \\ \langle a, \text{Trans}(20,50), 200 \rangle \langle b, \text{Trans}(42,50), 200 \rangle \langle \phi, \phi, 200 \rangle \rangle$$

A figura 8.8 ilustra como ficará a animação.

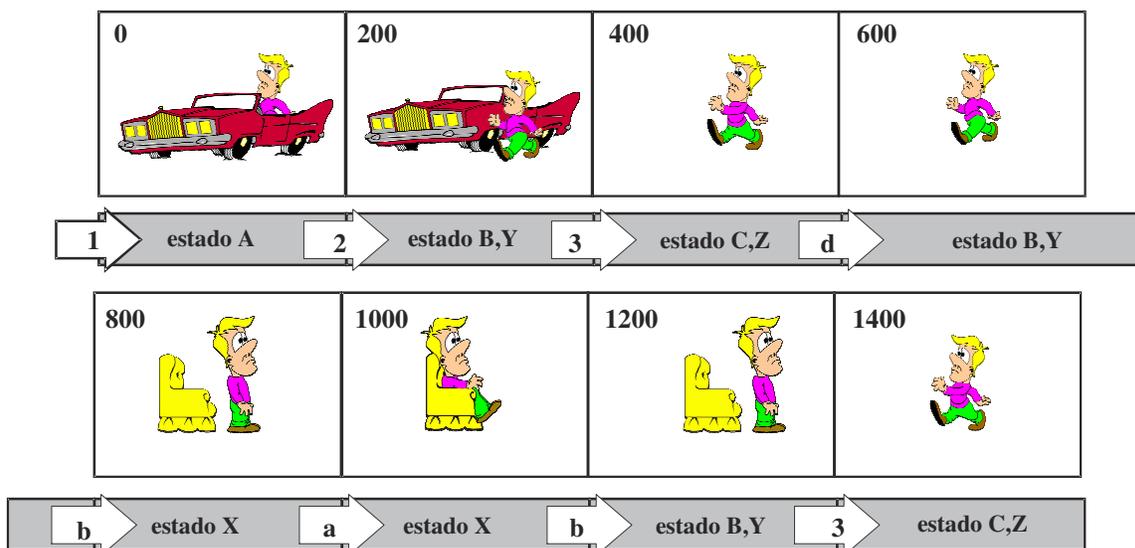


Figura 8.8: Animação obtida pela leitura da fita

8.9 Considerações Finais

Conforme descrito anteriormente o objetivo deste trabalho não é implementar as operações categoriais aqui estudadas e sim interpretá-las em animações computacionais.

Para cada uma das operações categoriais estudadas uma sugestão foi apresentada. Todas elas mantêm a mesma linha de raciocínio, ou seja, o que foi definido no AGA é mantido, o que se faz é mapear as fitas de entradas definidas em cada uma das operações categoriais para as fitas individuais de cada um dos atores envolvidos na animação.

Como isto é somente uma sugestão nada impede que uma nova maneira de implementação das operações categoriais seja encontrada e utilizada ao invés destas aqui exploradas.

9 ANIMAÇÃO OBTIDAS POR COMPOSIÇÃO DE OPERAÇÕES

9.1 Introdução

Este capítulo vem apresentar uma cena de uma possível futura animação, aparentemente complexa que podem ser criada utilizando-se as operações categoriais o que resulta em uma cena criada de forma simples, clara e de fácil implementação.

Como visto anteriormente nos capítulos 6 e 7, é possível que se faça composições de operações sobre operações quantas vezes se ache necessário ou até que se obtenha o resultado desejado.

Serão apresentados os atores que farão parte das cena e uma descrição da mesma. Para simplificação nem todos os autômatos resultantes de cada uma das operações categoriais utilizadas serão desenhados, só será ilustrado os pontos que são mais importantes para esclarecimento da operação aplicada.

9.2 Atores e Cena

Os atores que farão parte da animação foram criados baseados no modelo AGA e AGANd, eles são os seguintes:

Ator1 → **P** : O autômato do ator1 representa uma pessoa caminhando e está ilustrado pela figura 9.1,

Ator2 → **M** : O autômato do ator2 representa uma pessoa que estende a mão para cumprimentar e está representado pela figura 9.2.

Ator3 → **Chuta**: O autômato do ator3 representa uma pessoa chutando bola e está ilustrado pela figura 9.3.

Ator4 → **Goleiro**: O autômato do ator4 representa um goleiro e está ilustrado pela figura 9.4.

Ator5 → **Carro**: O autômato do ator5 representa um carro em estado normal e batido, está ilustrado pela figura 9.5.

Ator6 → **Caminhão**: O autômato do ator6 representa um caminhão normal e batido. Este autômato é o mesmo do ator5, ilustrado na figura 9.5, o que muda é a saída que será um caminhão normal no lugar do carro normal e um caminhão batido no lugar do carro batido, para simplificação o autômato não foi desenhado novamente.

Ator7 → **Sol**: O autômato do ator7 representa um sol e está ilustrado pela figura 9.6.

Ator8 → **Nublado**: O autômato do ator8 representa nublado. Este autômato é o mesmo do ator7, ilustrado na figura 9.6, o que muda é a saída que será uma nuvem, para simplificação o autômato não foi desenhado novamente.

Ator9 → Chuva: O autômato do ator9 representa chuva. Este autômato é o mesmo do ator7, ilustrado na figura 9.6, o que muda é a saída que será uma nuvem com pingos de chuva, para simplificação o autômato não foi desenhado novamente.

Ator10 → Avião: O autômato do ator10 representa um avião. Este autômato é o mesmo do ator7, ilustrado na figura 9.6, o que muda é a saída que será um avião, para simplificação o autômato não foi desenhado novamente.

Ator11 → Pássaro: O autômato do ator11 representa um pássaro voando. Este autômato é o mesmo do ator7, ilustrado na figura 9.6, o que muda é a saída que será um pássaro, para simplificação o autômato não foi desenhado novamente.

Ator12 → Fundo: O autômato do ator12 representa um fundo com Prédios, estrada, praça e campo de futebol. Este autômato é o mesmo do ator7, ilustrado na figura 9.6, o que muda é a saída que será o fundo, para simplificação o autômato não foi desenhado novamente.

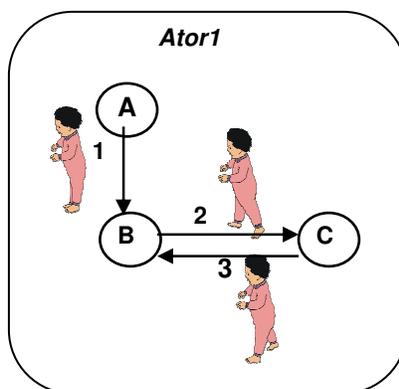


Figura 9.1: P (Pessoa caminhando)

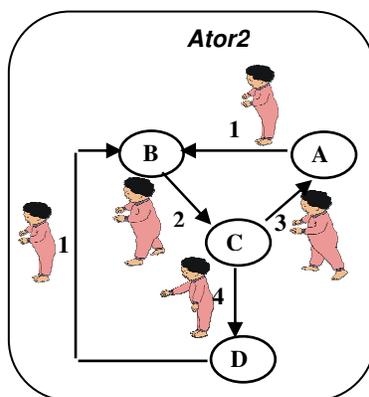


Figura 9.2: M (pessoa que estende a mão para cumprimentar)

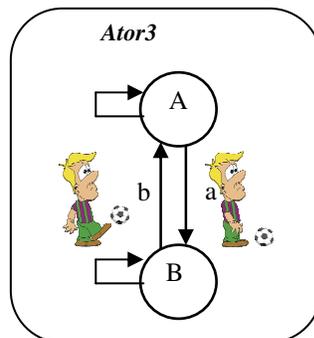


Figura 9.3: Chuta (Pessoa chutando bola)

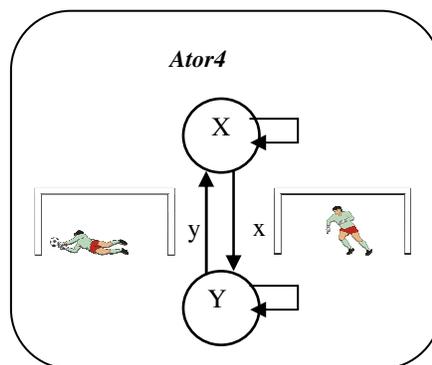


Figura 9.4: Goleiro

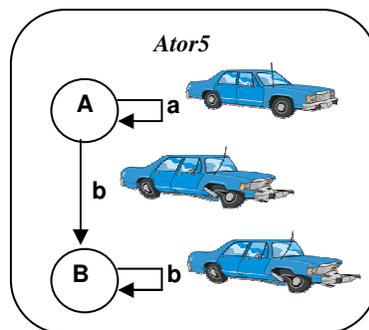


Figura 9.5: Carro

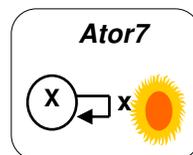


Figura 9.6: Sol

A cena desejada é a seguinte:

O fundo será: Prédios, em frente uma estrada, uma praça e um campinho de futebol no lado esquerdo da praça.

Se o dia estiver ensolarado ou nublado, aparecerão várias pessoas caminhando aleatoriamente na praça. Duas pessoas se encontram e se cumprimentam. Um avião ou um bando de pássaros estará sobrevoando os prédios. Em determinado momento aparecem dois homens jogando bola no campinho da praça. Um deles chuta a bola enquanto o outro faz o papel de goleiro.

Se o dia estiver com chuva, aparecerá um carro e um caminhão passando pela estrada, quando acontece um acidente envolvendo dois, aparecem alguns curiosos para ver o que está acontecendo.

9.3 Resolução

Aparentemente parece complexo montar estas cenas e num primeiro momento pensa-se em construir um novo autômato para que se obtenha o resultado desejado. Porém utilizando-se as operações categoriais investigadas neste trabalho é possível que se chegue a este resultado sem que se tenha que construir um novo autômato.

A resolução será explicada passo a passo para que o leitor consiga acompanhar o raciocínio, nenhum autômato resultante das operações categoriais utilizadas será desenhado, porque na verdade ele não é desenhado realmente, ele é obtido a partir das operações categoriais efetuadas.

Várias pessoas andando

Para que se consiga obter várias pessoas andando é necessário que se aplique a operação categorial de **produto**, várias vezes, sobre o ator1. Para que eles apareçam em diversas posições e em direções diferentes na tela de apresentação o projetista da fita de entrada tem que controlar as transformações e as posições dos atores, pois isto não é controlado pela operação categorial e sim na fita de entrada conforme definiu Accorsi.

A operação é a seguinte:

$$\text{pessoas} = (P \ x \dots x \ P)$$

Esta operação poderá ser aplicada até que se tenha o número de pessoas que se deseje na cena.

Dois pessoas se cumprimentando

Para que seja tenha os duas pessoas se cumprimentando é necessário que se aplique a operação categorial de **soma amalgamada sem a identificação das saídas** sobre o ator2, especificando quais as partes serão consideradas iguais. Para que um deles apareça do lado oposto o projetista da fita de entrada tem que aplicar uma transformação em um ator.

A operação é a seguinte:

$$\text{cumprimento} = (M \ +_{\text{aperto}} \ M)$$

Pessoas andando e duas delas se cumprimentando

Para que se tenha várias pessoas andando e duas deles se cumprimentando é necessário que se aplique a operação categorial de **produto** entre pessoas e cumprimento.

A operação categorial é a seguinte:

$$\text{andando_cumprimento} = \text{pessoas} \times \text{cumprimento}$$

Plano de Fundo com um dia ensolarado ou um dia nublado

Para que se tenha o Plano de Fundo com um dia ensolarado ou um dia nublado é necessário que se aplique a operação categorial de **coproducto** entre o sol e o nublado e ao resultado desta operação deverá ser aplicado o **produto** com o plano de fundo.

A operação é a seguinte:

$$\text{céu1} = \text{Fundo} \times (\text{Sol} + \text{Nublado})$$

A cena ficará conforme ilustra a figura 9.7.

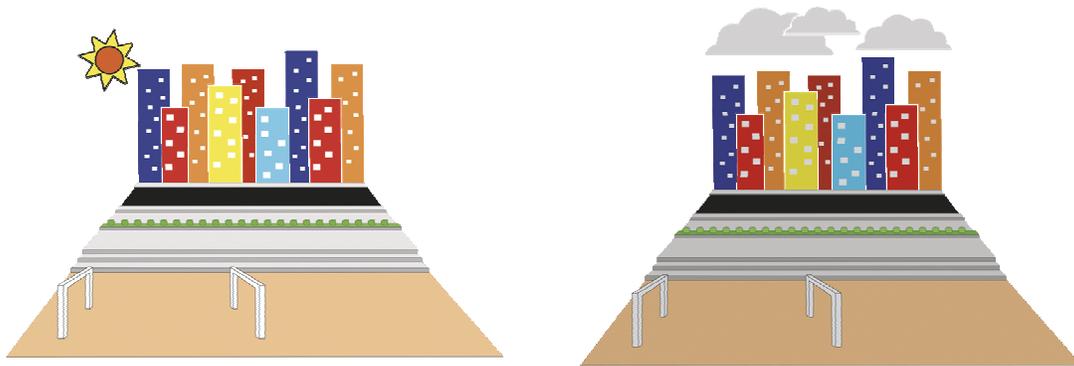


Figura 9.7 - Dia ensolarado e Dia nublado

Plano de Fundo com um dia ensolarado ou um dia nublado com um avião ou passáros voando

No dia nublado ou no dia ensolarado poderá aparecer um avião ou um pássaro voando. Para que se tenha este resultado é necessário que se aplique a operação categorial de **produto** entre céu1 com o **coproducto** do Avião e do Pássaro.

A operação é a seguinte:

$$\text{céu} = \text{céu1} \times (\text{Avião} + (\text{Pássaro}))$$

Pode-se desejar que se tenha mais de um pássaro no céu, então a operação seria modificada e ficaria da seguinte forma:

$$\text{céu} = \text{céu1} \times (\text{Avião} + (\text{Pássaro} \times \dots \times \text{Pássaro}))$$

A figura 9.8 ilustra como ficaria a cena.

Poderia ser efetuada a mesma operação com o avião, para que fosse obtido mais de um avião voando.

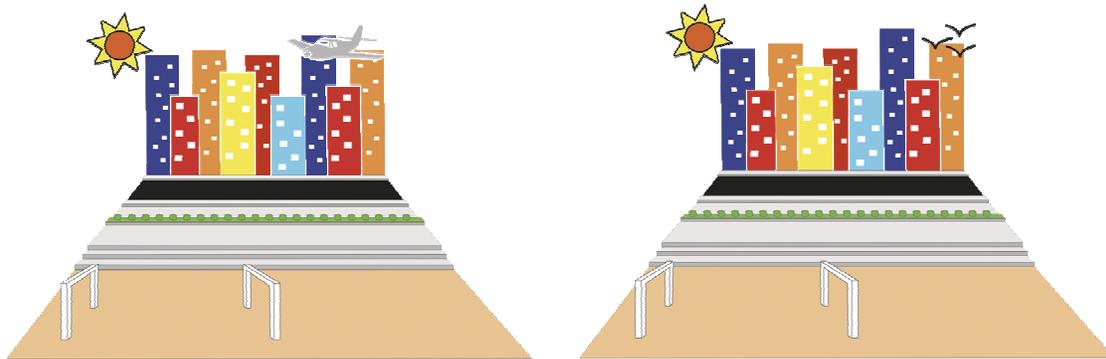


Figura 9.8 – Avião e pássaros voando

Jogo

Para que se tenha a animação de uma pessoa chutando bola e o goleiro defendendo basta aplicar a operação categorial de **produto** entre os atores chuta e goleiro. Porém o que se pretende é que só apareça a pessoa chutando a bola quando o goleiro estiver caído defendendo, e a pessoa com a bola em seus pés sem chutar quando o goleiro estiver de pé na frente da trave. Sendo assim torna-se necessário que se aplique a operação categorial de **Produto fibrado como sincronizador de sistemas** entre os atores chuta e goleiro, e que seja definida a tabela de sincronização.

Portanto, operação categorial é a seguinte:

Tabela de sincronização: jogador chutando a bola com o goleiro caído e o jogador parado e o goleiro de pé

$$\text{jogo} = \text{Chuta} \times \text{bolaGoleiro}$$

Pedaço da Cena

Para que se obtenha o seguinte pedaço da cena:

O fundo será: Prédios, em frente uma estrada, uma praça e um campinho de futebol no lado esquerdo da praça.

Se o dia estiver ensolarado ou nublado, aparecerão várias pessoas caminhando aleatoriamente na praça. Duas pessoas se encontram e se cumprimentam. Um avião ou um bando de pássaros estará sobrevoando os prédios. Em determinado momento aparecem dois homens jogando bola no campinho da praça. Um deles chuta a bola enquanto o outro faz o papel de goleiro.

A solução é a seguinte:

$$\text{Dia_sol_nublado} = \text{céu} \times \text{pessoas_cumprimento} \times \text{jogo}$$

Chuva

Para que se tenha um dia chuvoso é necessário que se aplique a operação de **produto** entre os atores plano de fundo e chuva.

A operação categorial é a seguinte:

$$\text{dia_chuva} = \text{Fundo} \times \text{Chuva}$$

Batida entre o carro e o caminhão

Para que se obtenha o carro e o caminhão batendo é necessário que se aplique a operação categorial de **produto** entre os atores carro e caminhão. Porém se a seguinte restrição for desejada:

Restrição: Na animação só deverá aparecer o caminhão em perfeito estado quando o carro também estiver em perfeito estado e o caminhão batido só deverá aparecer quando o carro também estiver batido.

Então para que se obtenha este resultado é necessário que se aplique a operação categorial de **produto fibrado com restrição de sistema** entre os atores carro e caminhão. Sendo assim a tabela de restrição deverá ser definida.

Portanto a operação categorial é a seguinte:

***Tabela de restrição:** Na animação só deverá aparecer o caminhão em perfeito estado quando o carro também estiver em perfeito estado e o caminhão batido só deverá aparecer quando o carro também estiver batido.*

$$\text{acidente} = \text{Caminhão} \times \text{batida Carro}$$

Pessoas curiosas

Para que se tenha as pessoas curiosas interessadas em ver o acidente é necessário que se aplique a operação categorial de **produto** entre o ator pessoa e o **produto** novamente com o acidente.

A operação categorial é a seguinte:

$$\text{curiosos} = \text{acidente} \times (P \times \dots \times P)$$

Pedaço da Cena

Para que se obtenha o seguinte pedaço da cena:

Se o dia estiver com chuva, aparecerá um carro e um caminhão passando pela estrada, quando acontece um acidente envolvendo dois, aparecem alguns curiosos para ver o que está acontecendo.

A solução é a seguinte:

$$\text{dia_chuva_acidente} = \text{dia_chuva} \times \text{curiosos}$$

A figura 9.9 ilustra este pedaço da cena.

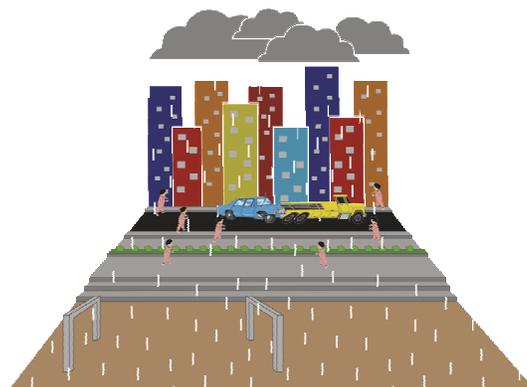


Figura 9.9 – Peça da cena

Cena completa

Para que se obtenha a cena inteira, proposta inicialmente, a solução é a seguinte:

$$\text{cena} = \text{dia_sol_nublado} + \text{dia_chuva_acidente}$$

Como foi dito inicialmente, uma cena muitas vezes considerada complexa pode ser criada utilizando-se operações categoriais de forma simples, clara e de fácil implementação.

Esta seria a operação categorial necessária para que se consiga a cena desejada, pode-se observar que foram utilizadas composições de operações e nenhum novo autômato foi construído. Os autômatos originais não foram perdidos em nenhum momento e foram reutilizados quantas vezes necessário. É importante esclarecer que no modelo AGA, o posicionamento dos atores na tela de exibição são definidos quadro a quadro e na própria fita de entrada. Portanto para que se evite sobreposições de imagens e que estas aparecem nas posições desejadas na tela de apresentação o projetista da fita deve definir as posições e as transformações específicas para cada ator em cada momento da animação.

9.4 Considerações Finais

Este capítulo, mostra de uma forma sucinta um exemplo do que é possível obter com a aplicação das operações categoriais, na realidade, ele resume todo a investigação sobre as operações categoriais feito por esta dissertação.

Para que se chegue a cena desejada, num primeiro momento se pensa que é necessário que se crie um novo autômato, e que provavelmente este novo autômato geraria uma explosão de estados. Porém nenhum novo autômato foi criado e a cena pode ser obtida utilizando-se as operações categoriais já investigadas e fundamentadas matematicamente.

Podê-se observar que as operações categoriais são poderosas o suficiente para que uma situação considerada complexa, de difícil resolução e implementação seja solucionada de forma simples e clara.

Uma das vantagens de se utilizar as operações categoriais sobre os autômatos dos atores de uma animação é que os atores originais não são perdidos, podendo assim serem reutilizados quantas vezes forem necessários. A noção de reuso é uma das principais características de Teoria das Categorias. O exemplo apresentado mostra isto com muita sutileza, porém outros exemplos envolvendo animações poderiam ser criados.

10 CONCLUSÃO E TRABALHOS FUTUROS

Esta última seção resume as conclusões obtidas em cada um dos capítulos anteriores. Também são sugeridos trabalhos futuros que poderão ser de interesse para o aprofundamento do assunto e aproveitamento dos conceitos criados.

10.1 Conclusões

Essa dissertação faz parte do Projeto *Hyper Seed* que agrega uma grande quantidade de pesquisas e trabalhos, todos desenvolvidos em uma mesma linha de raciocínio que é a de desenvolver uma fundamentação matemática para a unificação, de maneira coerente e (matematicamente) rigorosa, de especificações de textos e animações, baseada na Teoria dos Autômatos e das Categorias. É nessa linha que esta dissertação também está incluída, pois o que se pretende é utilizar a Teoria das Categorias como fundamento matemático para que se possa fazer composições entre os atores originais de uma animação a fim de se conseguir uma nova animação sem que se perca a animação original, utilizando a noção de reuso de partes ou até mesmo do todo de um ator.

No capítulo 3 foi feito um estudo sobre as animações por computador. Vários formatos de animação são descritos como o Macromédia Flash (SWF), o Scalable Vector Graphics (SVG), o Graphics Interchange Format (GIF), o Moving Picture Expert Group (MPEG-1) e o QuickTime e Audio Video Interleave (AVI), sendo que cada um possui um fim específico. O modelo AGA utiliza a idéia de quadro a quadro e uma animação bidimensional. É importante ressaltar que este modelo foi o utilizado para a aplicação das operações categorias abordadas nesta dissertação. O capítulo 4 descreve resumidamente o modelo AGA, este modelo é baseado em autômatos finitos determinísticos e possui características que favorecem sua aplicação na especificação e controle de animações para a *Web*. O encapsulamento das propriedades estéticas e comportamentais dos atores em uma unidade básica, o ator AGA, favorece que estes sejam reutilizados em diferentes animações. As variações comportamentais desses atores em cada animação podem ser exploradas simplesmente pela troca da fita de entrada. Essas características também colaboram para que animações novas sejam formadas a partir da combinação de animações existentes. Porém no modelo AGA não é explicado de forma clara, como se pode criar novas animações a partir das 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. No AGA é definido como se faz combinação entre os atores para que a partir daí se consigam animações distintas com a modificação das fitas de entrada de cada um dos atores envolvidos na animação, porém esta combinação não está

fundamentada matematicamente, é interessante lembrar que este modelo utiliza somente autômatos finitos determinísticos. No capítulo 5 foi apresentada uma sugestão para que se obtenham animações mais realistas, a utilização de autômatos finitos não determinísticos (AGANd), quando necessário, na criação dos atores da animação. Utilizando-se, na construção de atores, o modelo AGANd, é possível compor animações de forma mais realística e muito mais flexíveis, conseguindo-se assim animações que tenham um comportamento mais próximo com o que ocorre no mundo real. Para solucionar o não determinismo é sugerido a utilização de um AFNDp (Autômatos Finitos Não determinísticos com pesos), são atribuídos pesos para as arestas dos autômatos, esta idéia é baseada nos princípios Cadeias de Markov e dos Autômatos Estocásticos. O AFNDp aumenta o poder de expressão dos autômatos finitos não determinísticos. Na semântica proposta a máquina escolhe aleatoriamente o caminho a seguir a partir dos pesos especificados pelo projetista da mesma, que é responsável pela definição da prioridade de cada um dos caminhos. No capítulo 6 foram tratados as aplicações das operações categoriais de produto e coproduto entre os atores de uma animação gráfica. Pôde-se observar que tanto no Produto como no Coproduto o objeto resultante é sempre um novo ator a partir dos atores originais.

O Capítulo 7 tratou da aplicação das operações categoriais de Produto Fibrado e Soma Amalgamada sobre os atores de uma animação.

Pôde-se observar que tanto no Produto Fibrado como na Soma Amalgamada o objeto resultante é sempre um novo ator gerado a partir dos atores originais, portanto se o autômato original for modificado o resultado das operações será alterado automaticamente.

Com a definição das fitas de entrada para cada uma das operações é possível que se obtenha uma nova animação a partir dos atores originais. No estudo do Produto Fibrado foram verificados os resultados sob duas interpretações Produto Fibrado como restrição de um sistema e Produto Fibrado como sincronização, qualquer uma dessas interpretações poderá ser utilizada. No estudo da Soma Amalgamada também foram estudadas duas interpretações, Soma Amalgamada com a identificação das saídas e a outra sem a identificação das saídas.

Os autômatos resultantes da aplicação destas operações categoriais foram desenhados para efeito de ilustração, na realidade eles não são definidos ou desenhados, eles são o resultado das operações matemáticas.

No capítulo 8 foi sugerido uma forma de implementar todas as operações estudadas, estendendo dessa forma o protótipo funcional do AGA.

O texto mostra ainda que é possível fazer composições de operações categoriais. Nos capítulos 6 e 7 pequenos exemplos foram apresentados. No capítulo 9, um exemplo envolvendo todas as operações categoriais, estudadas nesta dissertação, foi mostrado. Este capítulo, na realidade, resume todo o estudo feito por esta dissertação. Para que se chegue a uma cena desejada, num primeiro momento se pensa em criar um novo autômato que provavelmente geraria uma explosão de estados. Porém nenhum novo autômato foi criado e a animação pode ser gerada baseada nas operações categoriais ditas simples e com isto está fundamentado matematicamente.

Pode-se observar que as operações categoriais são poderosas o suficiente para que uma situação considerada complexa, de difícil resolução e implementação seja solucionada de forma simples e clara.

Esta dissertação não trata da aplicação das operações categoriais sobre as fitas de entrada das animações, este é um tema que poderá ser explorado em um trabalho futuro.

O objetivo desta dissertação não é provar que estas operações categoriais são as melhores ou as piores soluções encontradas para se obter combinações entre os atores de uma animação e com a utilização das fitas de entrada definidas para cada uma das operações se consegue da melhor ou pior maneira as animações. O que se investigou aqui é como esses resultados foram obtidos com uma fundamentação matemática.

10.2 Trabalhos Futuros

Os trabalhos vislumbrados para continuar esta pesquisa estão dentro do contexto do projeto *Hyper Seed*, que está descrito com maiores detalhes no Anexo A.

- a) Criação de uma interface para a utilização das operações categoriais estudadas, no protótipo do modelo AGA e em outros softwares a serem desenvolvidos pelo grupo, em que as operações categoriais serão utilizadas.
- b) As operações categoriais estudadas nesta dissertação fazem parte da implementação de um software para deficientes utilizando-se dessas operações para compor as páginas da Web. Esta implementação é fruto de um Projeto enviado ao CNPq com o nome de PNEE - Autômato: Sistema Hipermídia para Portadores de Necessidades de Educação Especial, baseado em Teoria dos Autômatos. A estrutura teórica que será trabalhada neste projeto está bem fundamentada na teoria dos autômatos e em teoria das categorias. O uso de teoria dos autômatos para definir hipermídia fornece uma estrutura matematicamente bem definida e flexível. Teoria das categorias possui um conjunto de conceitos poderosos para a especificação de transformações sobre elementos hipermídia. O projeto na íntegra está disponível no Anexo3.
- c) Elaborar e aplicar um estudo de caso que consiga investigar a lógica que as pessoas utilizam para criar as animações. Este estudo deverá ser aplicado a pessoas de todas as faixas etárias, inclusive crianças. Sendo que algumas destas devem ter conhecimento de computação e outras não. Na elaboração do estudo deve-se tomar o cuidado para não utilizar termos técnicos como autômatos, Soma Amalgamada, Produto, Coproduto, entre outros. É necessário que se utilize uma linguagem de fácil entendimento sendo que o mesmo será aplicado a todas as faixas etárias e a pessoas que tenham e não tenham conhecimento de computação. O objetivo é que todos consigam criar suas próprias animações utilizando os autômatos e aplicando as operações categoriais que achem necessário.
- c) Implementação de todas as construções categoriais estudadas nessa dissertação.
- d) Estudo de outras operações categoriais aplicadas a animação gráfica e em outros projetos estudados pelo grupo.
- e) Aplicação das operações categoriais sobre as fitas de entrada dos autômatos.
- f) Estudar os pesos das arestas dos autômatos, no caso de não-determinismo, nas operações categoriais

10.3 Produção Científica

Esta dissertação de mestrado resultou na publicação do artigo relacionado abaixo até o momento do fechamento deste texto:

. VIEIRA, Christine; MENEZES, Paulo Fernando Blauth; NEDEL, Luciana Pocher Nedel. **Autômato finito não-determinístico com pesos aplicado a animações baseadas**

em autômatos finitos. Revista de Tecnologia e Ambiente v. 8, n2, jul./dez. 2002 da Universidade do Extremo Sul Catarinense, UNESC. Que está disponível na íntegra no Anexo B.

REFERÊNCIAS

- [AHO 95] AHO, A. V.; SETHI, R.; ULLMAN, J. **Compiladores. Princípios, Técnicas e Ferramentas**. Tradução: Daniel de Ariosto Pinto. Rio de Janeiro: Livros Técnicos e Científicos, 1995.
- [ACC 2000] ACCORSI, F.; TOSACNI, L. V. , MENEZES, P. B. Animação gráfica baseada na teoria de autômatos. In: SBES WORKSHOP, 2000, João Pessoa, **Anais...** João Pessoa: CEFET – PB, 2000. p. 122-127.
- [ACC 2001] ACCORSI, F.; MENEZES, P. B.; NEDEL, L. P. Animação gráfica baseada em autômatos temporizados sincronizados. In: WORKSHOP ON FORMAL METHODS, 4.; 2001, Rio de Janeiro. **Proceedings...** [S.l.]: SBC, 2001. p.75-80.
- [ACC 2002] ACCORSI, F.; MENEZES, P. B.; NEDEL, L. P. **Animação Bidimensional para World Wide Web Baseada em Autômatos Finitos**. 2002. 113p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [ALU 94] ALUR, Rajeev; DILL, David L. A theory of timed automata. **Theoretical Computer Science**, Amsterdam, v.126, p. 183-235, 1994.
- [APP 2000] APPLE COMPUTER INC. **QuickTime File Format**. Disponível em: <<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/pdf/frame.htm>>. Acesso em: dez. 2000.
- [CAM 2001] CAMPANI, C. A. P.; MENEZES, P. B. Characterizing the Software Development Process: a new approach based on Kolmogorov Complexity. In: INTERNATIONAL CONFERENCE ON COMPUTER AIDED SYSTEM THEORY AND TECHNOLOGY, 8., 2001, Las Palmas de Gran Canaria. **Formal methods and tools for computer science: extended abstract**. [Las Palmas de Gran Canaria: IUCTC, 2001]. P. 78-81.
- [DRO 89] DROBOT, V. **Formal Languages and Automata Theory**. New York: Computer Science Press, 1989.

- [FOL 90] FOLEY, J. et. al. **Computer Graphics: principles and practice**. USA: Addison-Wesley, 1990. 1174 p.
- [GRA 2003] GRANDI, R. H. **Astrha – Um Ambiente Gráfico e Interativo para Internet Baseado em Hiper-Animações e na Teoria dos Autômatos**. 2003. 244p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [HOP 69] HOPCROFT, J. E. ; ULLMAN, J. D. **Formal Languages and Their Relation to Automata**. Reading: Addison Wesley, 1969.
- [HOP 79] HOPCROFT, J. E.; ULLMAN, J. D. **Introduction to Automata Theory, Languages and Computation**. Reading: Addison-Wesley, 1979.
- [INT 96] INTERNATIONAL ORGANISATION FOR STANDARDISATION. **Short MPEG-1 description**. 1996. Disponível em: <<http://mpeg.telecomitalia.com/standards/mpeg-1/mpeg-1.htm>>. Acesso em: ago. 2002.
- [JAV 2002] JAVA.SUN.COM. **The source for Java(TM) Technology**. Disponível em: <<http://java.sun.com>>. Acesso em: ago. 2002.
- [JDO 2002] JDOM ORG. **JDOM**. Disponível em: <<http://www.jdom.org>>. Acesso em: ago. 2002.
- [KEM 60] KEMENY, J. G.; SNELL, J. L. **Finite Markov Chains**. Princeton, N.J.: D. Van Nostrand, 1960.
- [LEE 98] LEE, G. S. **A Classification of File Formats for Animation**. Disponível em: <<http://www.cs.ubc.ca/labs/imager/tr/ps/lee>>.1998.ps.gz. Acesso em: jan. 2003.
- [LEW 2000] LEW, M. S. Next-Generation Web Searches for Visual Content. **Computer**, [S.l.], v.33, n.11, p.46-53, Nov. 2000.
- [MAC 71] MAC LANE, S. **Categories for the working mathematician**. New York: Springer-Verlag, 1971.
- [MAH 2000] MACHADO, J. H. de A. P. **Hyper-Automaton: hipertextos e cursos na Web usando autômatos finitos com saída**. 2000. 149p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [MAD 2002] MACHADO, C.C. **XHA: eXtensible Hyper-Automaton**. 2002. 142p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

- [MAM 2002] MACROMEDIA. **Macromedia Flash Application Development Center**. Disponível em: <<http://www.macromedia.com/desdev/mx/flash/>>. Acesso em: ago. 2002.
- [MAH 2002] MAGALHÃES, G. de C. **Aga Player: Animação 2D Baseada em Autômatos para Web**. 2002. Projeto de diplomação (Bacharelado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [MAR 99] MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport Livros e Multimídia, 1999. p.221-301.
- [MEN 99] MENEZES, P. B.; MACHADO, J. P. Web Courses are Automata: a Categorical Framework. In: WORKSHOP DE MÉTODOS FORMAIS, 2., Florianópolis. **Anais...**[S.l.:s.n], 1999. p.79-88.
- [MEN 2000a] MENEZES, P. B. **Linguagens Formais e Autômatos**. 5. ed. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto, 2000.
- [MEN 2000b] MENEZES, P. B.; MACHADO, J. H. A. P. Hyper-Automaton: hypertext framework with categorial operation. In: SIMPÓSIO BRASILEIRO DE LINGUAGENS DE PROGRAMAÇÃO, 4., 2000. **Anais...** Recife: Centro de Informática da UFPE, 2000. p. 29-47.
- [MEN 2002] MENEZES, P. B.; HEUSLER, H. **Teoria das Categorias e Ciência da Computação**. Porto Alegre: Sagra Luzzatto, 2002.
- [MEN 2003] MENEZES, P. B. et al. **Laboratorio de Fundamentos da Computação**. Instituto de Informática da Universidade Federal do Rio Grande do Sul. Disponível em: <<http://teia.inf.ufrgs.br>> Acesso em: nov. 2003.
- [MIC 2002] MICROSOFT CORPORATION. **AVI RIFF file reference**. Disponível em: <http://msdn.microsoft.com/library/en-us/wcedshow/html/dxce_dshow_avi_riff_file_reference.asp>. Acesso em: set. 2002.
- [MOR 2002] MORAIS, C. T. Q. de. **Hyper-Automaton: Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos**. 2002. 111p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [PIE 91] PIERCE, B. **Basic Category Theory for Computer Scientists**. Cambridge, MA: The MIT Press, 1991.
- [PUE 88] PUEYO, X.; TOST, D. A Survey of Computer Animation. **Computer Graphics Forum**, Amsterdam, v.7, p.281-300, 1988.

- [SAI 2001] SAITO, D. S. Visualização de Grafos Aplicada a Autômatos Finitos com saída. In: SEMANA ACADÊMICA DO PPGC, 6., 2001, Porto Alegre.
- [SEB 2001] SEBESTA, R. W. **Concepts of Programming Languages**. 5th ed. Boston: Addison- Wesley, 2001.
- [STA 2002] STANFORD UNIVERSITY. **Stanford Encyclopedia of Philosophy**. Disponível em: <<http://plato.stanford.edu/archives/fall1997/entries/category-theory>> Acesso: set. 2002
- [STO 77] STOY, J. E. **Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory**. [S.l.]: MIT Press, 1977.
- [THA 85] THALMANN, N. M.; THALMANN, D. **Computer Animation: Theory and Practice**. Tokyo: Springer-Verlag, 1985. 239p.
- [THA 91] THALMANN, N. M.; THALMANN, D. **New Trends in Animation and Visualization**. New York: John Wiley, 1991.
- [W3Ca 2002] WORLD WIDE WEB CONSORTIUM. **Scalable Vector Graphics**. Disponível em: <<http://www.w3.org/Graphics/SVG/Overview.htm8>> Acesso em: ago. 2002.
- [W3Cb 2002] WORLD WIDE WEB CONSORTIUM. **W3C Synchronized Multimedia Home Page**. Disponível em: <<http://www.w3.org/AudioVideo/>> Acesso em: set. 2002.
- [WIK 2002] WIKIPEDIA. **The Free Encyclopedia**. Disponível em: <<http://www.wikipedia.com>> Acesso em: dez. 2002.
- [WIN 96] WINSKEL, G.; SASSONE, V.; NIELSEN, M. Models for concurrency: towards a classification. **Theoretical Computer Science**, Amsterdam, v. 170, n. 1-2, p. 297-348, 1996.
- [ZEL 85] ZELTZER, D. Towards an integrated view of 3-D computer animation. **The Visual Computer**, [S.l.], v.1, n.4, p. 249-259, 1985.

ANEXO A PROJETO HYPER SEED

Conselho Nacional de Desenvolvimento Científico e Tecnológico
- CNPq

Dados do Projeto e do(a) Proponente

Sigla:	Hyper Seed
Título do Projeto:	Hyper Seed - Framework, Ferramentas e Métodos para Sistemas Hipermídia voltados para EAD via WWW
Referência da Chamada:	2ª Fase da Chamada Conjunta MCT/SEPIN – FINEP – CNPq 01/2002, Programa de Apoio à Pesquisa e Desenvolvimento e Inovação em Tecnologia da Informação – PDI-TI, financiados pelo CNPq
Linha(s) de atuação predominante(*):	<input checked="" type="checkbox"/> (7) pesquisa e desenvolvimento <input type="checkbox"/> () projeto de demonstração <input type="checkbox"/> (5) inovação tecnológica <input type="checkbox"/> (3) transferência de tecnologia <input type="checkbox"/> (4) formação e capacitação de talentos humanos <input type="checkbox"/> (6) aplicações-chave <input type="checkbox"/> () estudo prospectivo
Coordenador do Projeto: E-mail:	Paulo Fernando Blauth Menezes blauth@inf.ufrgs.br
Instituição/Unidade Executora:	Instituto de Informática - Universidade Federal do Rio Grande do Sul
Data:	Porto Alegre, 12 de novembro de 2002

Palavras-chave: [www](#), [hipermídia](#), [mídia adaptativa](#), [teoria dos autômatos](#), [teoria das categorias](#), [ensino a distância](#)

1. Identificação e caracterização do problema (máximo de 1 página)

Com o desenvolvimento atual da Internet, começa a haver uma grande demanda por serviços via web. Uma das aplicações que pode tirar mais proveito das melhorias na capacidade de transferência de arquivos da internet e das novas mídias é a de ensino a distância [36]. Na verdade, algumas das críticas que se fazia ao ensino a distância, baseado tradicionalmente em correio convencional, rádio e televisão, em parte desapareceu com o surgimento do ensino a distância baseado em redes de computadores.

A partir da migração dos sistemas hipermídia em direção a Web, alguns autores perceberam que o novo ambiente não oferecia as condições necessárias para a criação de ambientes integrados de ensino. Como apontou [17], é preciso definir-se um sistema mais poderoso de WWW que seja capaz de estruturar o material hipermídia em módulos, que possam ser reutilizados em diversos contextos.

Assim, neste Projeto busca-se uma fundamentação matemática para a unificação, de maneira coerente e rigorosa, de especificações de hiperdocumentos e animações, e, baseado nesta fundamentação, desenvolver um protótipo e um conjunto de ferramentas para suporte ao desenvolvimento de conteúdo hipermídia/multimídia que permita uma integração natural a ambientes voltados para o Ensino a Distância (EAD) via WWW.

O modelo de hipermídia [30] a ser desenvolvido é baseado na representação da estrutura de uma hiperbase através de formalismos da Teoria de Autômatos [12]. Este modelo baseia-se no fato de que autômatos não somente capturam o poder descritivo dos grafos diretos, conhecidos por serem um abstração útil em sistemas de hipertexto [4], como também fornecem uma máquina abstrata para o controle e análise de hipertextos e animações, além de ser um formalismo universalmente aceito e possuir uma história de uso em muitas disciplinas como uma técnica de diagramação formal [6, 21]. Os autômatos podem ser representados por uma estrutura de grafo, a qual pode ser manipulada sobre vários aspectos. Grafos com uma semântica formal podem ser usados para prover interfaces de programação para o controle de material hipermídia. Diversos trabalhos relacionados ao uso de representações gráficas na modelagem de aplicações hipermídia utilizam outras construções, como Mapas Conceituais [9], Workflow [2], modelos Entidade-Relacionamento, Orientação a Objetos [10], e também Redes de Petri (o modelo Trellis [8] é um exemplo).

Adicionalmente, utilizando-se as ferramentas da Teoria das Categorias [15, 20] é possível definir precisamente um conjunto de operações composicionais sobre os componentes de uma hiperbase (hiperdocumento e animações), ou seja, construtores capazes de gerar novos elementos hipermídia sobre outros já existentes. Tal aplicação permitirá uma maior flexibilidade na construção e manipulação de elementos hipermídia na WWW.

2. Justificativa (máximo de 1 página)

A educação científica e de qualidade capaz de sustentar o avanço tecnológico é sempre um objetivo a ser buscado por qualquer nação. Neste contexto, o advento do e-learning gerou grandes expectativas da popularização do ensino científico de qualidade. No entanto, o e-learning, na maioria dos casos, é uma simples transposição de conteúdos tradicionais (livros, revistas, etc) para Internet. Ou seja, as reais potencialidades ainda estão

por serem exploradas pela média dos professores e empresas, principalmente se comparado aos meios de aprendizado (autodidatas/apoio) tradicionais.

É típico da Computação que qualquer solução gerada para abordar questões como esta será temporária, pois a dinâmica do surgimento de novas tecnologias necessariamente devem ser incorporadas com rapidez. Neste caso, freqüentemente a solução perde grande parte das suas qualidades originais, resultando em sistemas "inchados".

Historicamente, um dos principais objetivos a ser buscado na utilização dos fundamentos matemáticos na computação é o de inspirar sistemas melhores, mais simples, com implementações mais eficientes e robustas [35].

Assim, uma das premissas do Sistema Hyper Seed é desenvolver uma fundamentação matemática para a unificação, de maneira coerente e (matematicamente) rigorosa, de especificações (baseadas na Teoria dos Autômatos e Teoria das Categorias) de hiperdocumentos e animações, e, baseado nesta fundamentação, desenvolver um protótipo para suporte ao desenvolvimento de conteúdo hipermídia/multimídia que permita uma integração natural a ambientes voltados para o EAD via WWW.

O sistema HyperSeed permitirá que o conteúdo disposto em uma base de dados seja apresentado em diversos níveis de amplitude e profundidade. Isto permitirá que disciplinas de diversos cursos possam dispor de uma base de conteúdos avançados para o desenvolvimento de disciplinas interativas, dirigidas pela demanda dos alunos e baseadas na construção do conhecimento.

Como resultado, é esperado que tais fundamentos e aplicação-chave contribuam para o avanço do estado-da-arte em desenvolvimento e formalização de sistemas hipermídia e induzam novas ferramentas e métodos que, quando usados de forma apropriada, reduzirão o custo e esforço de manutenção e desenvolvimento de bases de dados voltadas ao ensino a distância, bem como a disponibilização de material hipermídia e multimídia via WWW.

3. Objetivos e metas (máximo de 2 páginas)

a) Objetivos

Os objetivos deste projeto são:

1. Baseado nos modelos formais desenvolvidos e nos resultados atingidos em três projetos de pesquisa desenvolvidos pelo Grupo
 - (a) Hyper-Automaton, um sistema adaptativo de disponibilização de conteúdo hipertexto;
 - (b) AGA, Animação 2D Baseada em Autômatos para a Web;
 - (c) Nautilus, linguagem de especificação Orientada a Objetos, com facilidades não-tradicionais inspiradas em Teoria das Categorias;
 desenvolver uma fundamentação matemática para a unificação, de maneira coerente e (matematicamente) rigorosa, de especificações de textos e animações, baseada na Teoria dos Autômatos e das Categorias.
2. Baseando-se em tais fundamentos, desenvolver um protótipo de sistema com suporte ao desenvolvimento de conteúdo multimídia e hipermídia com ênfase em aplicações de e-learning.
3. Validar este protótipo em dois contextos:
 - (a) Num contexto típico de instituição de ensino, junto ao Instituto de Informática da UFRGS;

- (b) Num contexto corporativo, junto à empresa PLANCTA (www.plancta.com.br), em experiências de massa, estimado em pelo menos 15.000 usuários.

b) Metas

Ano 1: Desenvolvimento do framework matemático e especificação do protótipo.

Ano 2, semestre 1: Implementação do protótipo da aplicação-chave e definição dos estudos de caso.

Ano 2, semestre 2: Instanciação e validação dos estudos de caso.

Ano 3, semestre 1: Formalização e validação final do modelo e aplicação-chave desenvolvidos, e disseminação da aplicação-chave para outras instituições.

4. Relevância dos resultados e os impactos esperados (máximo de 2 páginas)

a) Indicadores de resultados (ao final do projeto)

Pretende-se ao final do projeto ter produzido e publicado um bom número de artigos científicos que apontem para a corroboração dos resultados teóricos na comunidade científica internacional e nacional. Artigos descrevendo a aplicabilidade do protótipo de ferramenta a ser construído e os diversos estudos de caso validarão a sua adequação como proposta para seu posterior desenvolvimento comercial. Estes últimos descreverão a forma com que cada estudo de caso foi conduzido e seu uso pela ferramenta. Um relatório final fechará o projeto no que diz respeito a produção acadêmica.

O desenvolvimento deste projeto, principalmente na área teórico-formal, será fortemente centrado na formação de doutores, mestres e graduados em Ciência da Computação com forte apoio de bolsistas de DTI e IC. Como trata-se de um projeto que em diversos momentos é multidisciplinar, será dada continuidade à política correntemente utilizada pelo Grupo de colaboração e/ou co-orientação com outros grupos complementares, tanto da UFRGS quanto de outras instituições.

O produto da aplicação chave constitui-se em um sistema para composição e manipulação de hiperdocumentos na Web, fundamentado em conceitos teórico-formais, com destaque para a Teoria dos Autômatos e a Teoria das Categorias, incluindo um modelo para representação de animações gráficas que proporciona a estruturação do conteúdo multimídia. Para o sistema Hyper Seed será desenvolvido um manual do usuário e de instalação do software. Este manual apresentará ainda a metodologia a ser utilizada para o desenvolvimento de material hiperímídia sobre o sistema apresentado.

b) Indicadores de progresso (ao final de cada 12 meses de projeto)

Indicadores de progresso se darão a partir dos workshops internos de avaliação de resultados. Nestes workshops os integrantes do projeto farão relatórios de atividades de condução da pesquisa/desenvolvimento a seu cargo. Após cada workshop um relatório conjunto, com anuência de todos os integrantes do grupo será redigido, não só como uma forma de tornar todos os pontos claros a totalidade dos integrantes da pesquisa, mas também com o objetivo de firmar acordo nas etapas posteriores a serem realizadas. É claro que resultados de condução da pesquisa por cada integrante individualmente, ou na parcela

do grupo, devem e serão submetidos a foro internacional e/ou nacional especializado para corroboração e validação da pesquisa do grupo. Serão realizados um total de 3 workshops:

- O primeiro, ao término do Ano 1, onde será apresentado como resultado da etapa de pesquisa o framework matemático e a especificação do protótipo;
- O segundo, ao término do Ano 2, onde serão apresentados, além dos resultados da implementação do protótipo e validação dos estudos de caso, também metas de adequação do framework e do protótipo para a terceira e última etapa;
- O terceiro, ao término do primeiro semestre do Ano 3, onde será apresentado o relatório final de conclusão do Projeto.

A parte de implementação do protótipo de ferramenta terá a avaliação de progresso medida, além da submissão/publicação de artigos técnico-científicos, por relatório do grupo de usuários que conduzem os estudos de caso. Esperamos contar com a empresa PLANCTA para a condução desta etapa.

c) Repercussão e/ou impactos dos resultados

Conforme já introduzido, é esperado que os fundamentos teóricos e aplicação-chave contribuam para o avanço do estado-da-arte em desenvolvimento e formalização de sistemas hipermídia e induzam novas ferramentas e métodos que, quando usados de forma apropriada, reduzirão o custo e esforço de manutenção e desenvolvimento de bases de dados voltadas ao ensino a distância, bem como a disponibilização de material hipermídia e multimídia, via WWW.

Como principal produto teórico-formal, espera-se obter um framework matemático (baseado em Teoria dos Autômatos e Teoria das Categorias) capaz de unificar os três frameworks desenvolvidos nos projetos antecessores, suficientemente expressivo para tratar todas as questões aqui propostas.

Como principal produto aplicado, espera-se obter um sistema semi-automatizado de hipermídia e multimídia na WWW, que incorpore os resultados do modelo formal desenvolvido através de modernas técnicas de engenharia de software (com ênfase em orientação a objetos e composição de sistemas) a ser disponibilizado à sociedade através da caracterização do mesmo como software livre.

Espera-se que este trabalho contribua para que a comunidade de pesquisadores tenha condições de dominar o conjunto de tecnologias de aquisição, armazenamento, processamento, exibição, animação e distribuição de informação, englobando modernas tecnologias de manipulação de informações (mais notadamente a grande gama de conceitos e aplicações em torno do XML).

Desta forma, a expectativa é de uma forte interação com a indústria no segundo ano do projeto, a partir de estudos de caso a serem utilizados na validação da metodologia definida e a transferência de tecnologia para a empresa parceira.

5. Considerações finais (máximo de ½ página)

O grupo de pesquisa proponente possui experiência na formalização de modelos e desenvolvimento de sistemas para a WWW como demonstram algumas das publicações relevantes a este projeto obtidas por seus integrantes e listadas no item Referências

Bibliográficas (a lista completa das publicações pode ser obtida na Plataforma Lattes do CNPq). Observe-se que diversas publicações são em conjunto com outros grupos de pesquisa de instituições nacionais e internacionais.

O coordenador do projeto possui também forte atuação na formação de recursos humanos nesta área de pesquisa, incluindo a conclusão, no último ano, da orientação de três projetos de graduação e quatro trabalhos de mestrado, além de outros em andamento.

O principal produto já desenvolvido pelo grupo encontra-se na página <http://teia.inf.ufrgs.br> e se constitui em um sistema semi-automatizado para a organização de hiperdocumentos instrucionais e um *player* para animações baseadas em autômatos. Este sistema está atualmente sendo utilizado como forma de suporte ao ensino de disciplinas no curso de Ciência da Computação da UFRGS e pode ser encontrado no link <http://teia.inf.ufrgs.br>.

Adicionalmente, com a parceria proposta com uma empresa privada, este Projeto garante que grande parte da tecnologia desenvolvida possa ser testada e utilizada em contextos comerciais normais.

Finalmente, devemos lembrar o desafio que representa o trabalho proposto neste projeto. A área envolve uma grande quantidade de conhecimentos de áreas distintas, evidenciando-se assim um caracter interdisciplinar e multidisciplinar.

6. Referências bibliográficas

- [1] ACCORSI, F.; MENEZES, P. F. Blauth; NEDEL, L. P. Animação Gráfica Baseada em Autômatos Temporizados Sincronizados. In: WMF'2001: IV Workshop de Métodos Formais, 2001, Rio de Janeiro. Anais do IV Workshop de Métodos Formais. Rio de Janeiro: SBC, 2001. v. 1, p. 75-80.
- [2] ADELSBERGER, H.; KÖRNER, F. X.; PAWLOWSKI, J. M. A Conceptual Model for an Integrated Design of Computer Supported Learning Environments and Workflow Management Systems. In: XV IFIP World Computer Congress, 1998, Vienna. Proceedings of the XV IFIP World Computer Congress - Teleteaching 98. Vienna: IFIP, 1998. p.55-64.
- [3] CARNEIRO, C. R. J. B.; REIS, R. Q.; MENEZES, P. F. Blauth. Especificação Formal de uma Ferramenta de Trabalho Colaborativo através da Composição de Objetos Nautilus. In: SBES'99: XIII SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 1999, Florianópolis. Anais do XIII Simpósio Brasileiro de Engenharia de Software. Florianópolis: SBC, 1999. p.95-110.
- [4] CONKLIN, J. Hypertext: An Introduction and Survey. IEEE Computer, v.20, n.9, p.17-41, Set. 1987.
- [5] DIVERIO, T. A., MITO, I. V. Foundations for Virtual Enviroments to Support Theory of Computation Teaching In: Advances Educational Technologies: multimedia, www and distance education ed.New York : John Wiley & Sons, 2001
- [6] DIVERIO, T. A.; MENEZES, P. F. B. Teoria da Computação: máquinas universais e computabilidade.2 ed. Porto Alegre : Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2000, v.1. p.224.
- [7] FREITAS, C. M. D. S., CHUBACHI, O. M., LUZZARDI, P. R. G., CAVA, R. A. Introdução à Visualização de Informações. Revista de Informática Teórica e Aplicada. Porto Alegre, RS, v.8, n.2, p.143 - 158, 2001.
- [8] FURUTA, R.; STOTS, P. D. Programmable Browsing Semantics in Trellis. In: Proceeding of the Hypertext'89, 1989, p.27-42.
- [9] GAINES, B. R.; SHAW, M. L. G. Concept Maps as Hypermedia Components. 1996. <http://ksi.cpsc.ucalgary.ca/articles/ConceptMaps/>
- [10] GARZOTTO, F.; SCHWABE, D.; PAOLINI, P. HDM: A Model Based Approach to Hypertext Application Design. ACM Transactions on Information Systems, v.11, n.1, p.1-26, January 1993.
- [11] [HAE 2001] HAEUSLER, E. H.; MESEGUER, J. Mathematical and Engineering Foundations for Interoperability through Architecture. In: Proceedings of the Projects Evaluation Workshop CNPq/NSF/Inria, Ed. Nelson Prugner, Paulo Ernesto Castilho Lima e Celso Deusdeti Costa., p. 175-208, 2001.
- [12] HOPCROFT, J. E.; ULLMAN, J. D., Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.
- [13] MACHADO, C. C.; FEDERIZZI, G. L.; MENEZES, P. F. B. Flexibility and Adequacy of the Output's Layout of the Content Displayed in The Hyper-Automaton System. In: IC'2001: International Conference on Internet Computing, 2001, Las Vegas. 2nd International Conference on Internet Computing. Las Vegas: CSREA Press, 2001. v. 1, p. 424-430.

- [14] MACHADO, J. H. A. P.; MORAIS, C. T. Q. de; MENEZES, P. F. Blauth; REIS, R. A. da Luz. Structuring Web Course Pages as Automata: revising concepts. In: RIAO'2000: Recherche d'Informations Assistée par Ordinateur, 2000, Paris. Content-Based Multimedia Information Access: conference proceedings. Paris: C.I.D., C.A.S.I.S., 2000. v.1, p. 150-159.
- [15] MAC LANE, S. Categories for the Working Mathematician. Springer-Verlag, 1971.
- [16] MANSSOUR, I. H., FURUIE, S. S., OLABARRIAGA, S. D., FREITAS, C. M. D. S. Visualizing Inner Structures in Multimodal Volume Data In: Brazilian Symposium on Computer Graphics and Image Processing, 2002, Fortaleza. SIBGRAPi 2002. Los Alamitos: IEEE Computer Society, 2002. v.1. p.51-58.
- [17] MAURER, H. Necessary Ingredients of Integrated Network Based Learning Environments. In: ED-MEDIA/ED-TELECOM Educational Multimedia/Hypermedia and Telecommunications, Charlottesville, v.1, 1997, p.619-624.
- [18] MENEZES, P. F. B.; COSTA, S. A. da; MACHADO, J. H. A. P.; RAMOS, J. Nautilus: a concurrent anticipatory programming language. In: CASYS'2001: 5th International Conference on Computing Anticipatory Systems, 2001, Liège. AIP Conference Proceedings. Melville: American Institute of Physics, 2002. v. 627, p. 553-564.
- [19] MENEZES, P. F. B.; MACHADO, J. H. A. P. Adaptive Web Courses: a Categorical Framework International Journal of Computing Anticipatory Systems, Liege, v.9, p. 318-336, 2001.
- [20] MENEZES, P. F. B.; HAEUSLER, E. H. Teoria das Categorias para Ciência da Computação.1 ed. Porto Alegre : Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2001, v.1. p.324.
- [21] MENEZES, P. F. B. Linguagens Formais e Autômatos.4 ed. Porto Alegre : Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2001, v.1. p.192.
- [22] MENEZES, P. F. B.; MACHADO, J. H. A. P. Hyper-Automaton: hypertext framework with categorical operations. In: SBLP'2000: IV Simpósio Brasileiro de Linguagens de Programação, 2000, Recife. Anais do IV Simpósio Brasileiro de Linguagens de Programação. Recife: Centro de Informática da UFPE, 2000. p. 29-47.
- [23] MENEZES, P. F. B.; MACHADO, J. H. A. P. Web Courses are Automata: a categorical framework. In: WMF'99: II Workshop de Métodos Formais, 1999, Florianópolis. II Workshop on Formal Methods. Florianópolis: UFSC, 1999. p. 79-88.
- [24] MENEZES P. F. B. Compositional Reification of Concurrent, Interacting Systems. In: PDPTA'98: International Conference on Parallel and Distributed Processing Techniques and Applications, 1998, Las Vegas. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications. Las Vegas: CSREA Press, 1998, p. 1754-1761.
- [25] MENEZES, P. F. B., SERNADAS, A., COSTA, J.F., Nonsequential Automata Semantics for Concurrent, Object-Based Language. In First US-BRAZIL Workshop on Formal Foundations of Software Systems, 1998, Rio de Janeiro Electronic Notes in Theoretical Computer Science, Amsterdam Elsevier, 1998, v. 14.
- [26] MENEZES, P. F. B.; COSTA, J. F. Synchronization in Petri Nets, Fundamenta Informaticae, Amsterdam, v. 26, n.1, p. 11-22, 1996.
- [27] MORAIS, C. T. Q. de; MACHADO, J. H. A. P.; MENEZES, P. F. B.; REIS, R. A. da Luz. A Web Teaching System Based on Formal Methods. In: WCC'2000: 16th IFIP World Computer Congress, 2000, Beijing. Proceedings ICEUT, IFIP World Computer Congress. Beijing: PHEI, 2000. v. 1, p. 221-224.
- [28] NEDEL, L. P.; THALMANN, D. Anatomical modeling of deformable human bodies. The Visual Computer, Heidelber, Germany, v.16, n.6, p.306-321, 2000.
- [29] NEDEL, L. P.; FREITAS, C. M. D. S.; WAGNER, F. R. Simulação de objetos deformáveis baseada na análise dinâmica. Revista de Informática Teórica e Aplicada, Porto Alegre, v.5, n.2, p.23-33, 1999.
- [30] [NIELSEN 90] NIELSEN, J. Hypertext and Hypermedia. San Diego: Academic Press, 1990.
- [31] NITZKE, J. A., CARNEIRO, M. L. F., FRANCO, S. R. K. Ambientes de Aprendizagem Cooperativa Apoiados pelo Computador e sua epistemologia. Informática na Educação Teoria e Prática, Porto Alegre, v.5, n.1, p.13-23, 2002.
- [32] NITZKE, J. A., POLONIA, E., SLOCZINSKI, H., ZEVE, C., LIMA, J. V. A CD- ROM environment for collaborative learning integrated to the Internet In: Computers and Advanced TEchnology in Education, 2000, Cancun. Proceedings of the IASTED International Conference. Calgary: IASTED/ ACTA PRESS, 2000. p.297-302.
- [33] NITZKE, J. A., MANFROI, V., SILVEIRA, A. E. Give us paper notes - Problems and solutions for a new strategy to food engineers based on computer supported collaborative learning In: nternational Conference on Engineering and Computer Education, 2000, SÃO PAULO. IEEE - ICECE2000 - Proceedings. São Paulo: IEEE/ SENAC, 2000.
- [34] PASQUALOTTI, A., FREITAS, C. M. D. S. Experimentação de Ambiente Virtual para Melhoria do ensino-aprendizagem de Matemática. Boletim de Educação Matemática. UNESP - Rio Claro, São Paulo: , v.14, n.16, p.79 - 101, 2001.
- [35] [STO 77] STOY, J. E. Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory, MIT Press, 1977.
- [36] [WEB 98] WEBER, T. et al. Uma Experiência com Hiperdocumentos e Internet no Suporte a Disciplinas de Computação, VI WEI - Workshop sobre Ensino em Informática, vol. 1, pp. 532-545, Belo Horizonte, 1998.

ANEXO B ARTIGO PUBLICADO NA REVISTA CIENTÍFICA DA UNESC (UNIVERSIDADE DO EXTREMO SUL CATARINENSE)

AUTÔMATO FINITO NÃO-DETERMINÍSTICO COM PESOS APLICADO A ANIMAÇÕES BASEADAS EM AUTÔMATOS FINITOS

Non-deterministic finite automatum with weights applied to animation based on finite automata

Christine Vieira*

Paulo Blauth Menezes**

Luciana Pocher Nedel**

Resumo

Este artigo descreve um modelo de autômato finito, baseado na teoria de autômatos para representar um *Autômato Finito Não-Determinístico*. O modelo proposto, *AFNDp* (Autômato Finito Não-determinístico com pesos) aumenta o poder de expressão dos *Autômatos Finitos Não-Determinísticos*, utilizando a noção de pesos em suas arestas. A máquina escolhe aleatoriamente o caminho a seguir a partir dos pesos especificados pelo projetista da mesma, que é responsável pela definição da prioridade de cada um dos caminhos. A aplicação do *AFNDp* no modelo AGA (Animação Gráfica Baseada em Autômatos Finitos) tem como objetivo estender o modelo para que se possam criar animações que serão expressas de forma simples (menos estados e menos transições), o modelo originado é denominado de *AGA-Np* (Animação Gráfica Baseada em Autômatos Finitos Não-Determinísticos com pesos).

* Universidade do Extremo Sul Catarinense (UNESC). Endereço: Caixa Postal 3.167, 88.806-000, Criciúma – SC, Brasil. E-mail: cvi@unesc.rct-sc.br

** Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS). Endereço: Caixa Postal 15.064, 91.501-970, Porto Alegre – RS, Brasil. E-mail: (blauth,nedel)@inf.ufrgs.br.

Palavras-Chave: animação gráfica, autômato finito determinístico, autômato finito não-determinístico.

Abstract

This article describes a model of finite automatum based on the automata theory in order to represent a non-deterministic finite automatum. The proposed model, AFNDp (Non-deterministic Finite Automata with weights) increases the power of expression of the Non-deterministic Finite Automata, using the notion of weights in its edges. The machine randomly chooses the way to follow, based on the weights specified by its creator, who is responsible for determining the priority of every way. The application of AFNDp in the model AGA (Computational Animation Based of Finite Automata) aims at extending the model so that animations which will be expressed in a simple way (less states and less transitions) can be created, the originated model is called AGANp (Computational Animation Based on Non-deterministic Finite Automata with weights).

Keywords: computational animation, deterministic finite automata, non-deterministic finite automata.

Introdução

Uma das maneiras de se representar sistemas regulares, ou seja, sistemas que podem ser expressos por um formalismo regular, é mapear o seu comportamento através de Autômatos Finitos.

Um *Autômato Finito (AF)* é definido como sendo um conjunto finito de estados, no qual um q_0 é identificado como estado inicial e um subconjunto de estados (eventualmente um conjunto vazio) é considerado como estado final. (AHO 95; MEN 2000, HOP 79, DRO 89).

Uma das vantagens de se utilizar *autômatos finitos* é que sua simulação requer apenas uma operação para processar um símbolo de entrada e uma quantidade finita e fixa de memória para o processamento. Essas operações podem ser vistas como ciclos discretos do relógio em que cada entrada processada corresponde a um ciclo (GERSTING). Esses *AF* podem ser divididos em *Autômatos Finitos Determinístico (AFD)* e *Autômatos Finitos Não-determinístico (AFND)*.

Um *autômato finito determinístico* é aquele em que, no máximo, uma transição saindo de um estado pode ser possível para o mesmo símbolo de entrada. Um *autômato finito não-determinístico* é aquele em que pode existir mais de uma transição saindo de um estado para o mesmo símbolo de entrada (AHO, 1995). O *não-determinismo* é uma importante generalização dos modelos de máquinas, sendo de fundamental importância no estudo da Teoria da Computação e da Teoria das Linguagens Formais (MENEZES, 2000).

Com um *AFND*, é possível representar alguns sistemas de maneira simples, o que muitas vezes não se consegue com um *AFD*.

Este artigo define um novo modelo de representação de um *AFND*, o *AFNDp (Autômato Finito Não-Determinístico com pesos)*, que procura explorar as potencialidades de representação e implementação dos *AFND*, com ênfase para aplicação no modelo AGA (Animações para Web baseadas em Autômatos Finitos) (ACCORSI & MENEZES, 2000).

Na primeira parte será apresentada a definição de *não-determinismo interno e externo*, pois neste artigo será dada ênfase ao *não-determinismo interno*. A parte seguinte tratará da apresentação visual de animações baseadas em *AFND*. Logo depois, o modelo *AFNDp* será descrito e formalizado a partir da extensão de *autômatos finitos não-determinísticos*. A seguir, demonstra-se a sua aplicação em animações baseadas em *AF*, modelo AGA (ACCORSI & MENEZES, 2000). Como todos os modelos de *AF* para geração de animação no modelo AGA são baseados em *AFD*, este artigo demonstrará que tais animações poderão ser representadas por *AFNDp*, resultando no AGA-Np, o que aumentará o seu poder de expressão. Finalmente, serão apresentadas conclusões e sugestões para trabalhos futuros.

Não-determinismo interno e externo

Uma das vantagens de se utilizar *autômatos finitos não-determinísticos* é que ele pode ser muito menor em número de estados e transições do que o seu *autômato finito determinístico* equivalente (AHO et al., 1995).

Os *autômatos finitos não-determinísticos* podem ser classificados por seu *não-determinismo* em *interno* e *externo*. Entende-se por *não-determinismo interno* aquele em que a máquina escolhe o caminho a seguir e *não-determinismo externo* aquele em que um fator externo, por exemplo o usuário, escolhe o caminho a seguir.

Para exemplificar o *não-determinismo externo e interno*, pode-se tomar o caso de uma máquina de refrigerantes e seu usuário. A máquina de refrigerantes possui os seguintes botões: Guaraná (Botão 1), Coca-cola (Botão 2), Coca-cola (Botão 3) e duas saídas (Saída 1 e Saída 2). O usuário pode pressionar qualquer botão para obter o seu refrigerante, mas se ele escolher Coca-cola haverá duas opções, Botão 2 e 3. A essa escolha feita pelo usuário chamamos de *não-determinismo externo*, em que algum fator externo, independente da máquina, decide o caminho a seguir. Por sua vez a máquina, de forma aleatória, escolherá a saída em que liberará o refrigerante. Nesse caso, o usuário não tem nenhuma influência na escolha da máquina, ela é que define o caminho a seguir, a isso chamamos de *não-determinismo interno*.

O modelo proposto neste artigo baseia-se no *não-determinismo interno*, mas cabe ao projetista da máquina definir pesos aos caminhos do autômato, desta forma a máquina continua escolhendo o caminho, aleatoriamente, a partir dos pesos definidos pelo projetista.

Apresentação visual de Animações baseadas em Autômatos Finitos Não-determinísticos.

A partir das definições de *não-determinismo interno e externo*, o autômato sempre segue um dos caminhos, seja ele definido internamente e de forma aleatória pela máquina ou externamente pelo usuário. Se isso fosse aplicado a uma animação baseada em autômato finito, a animação exibida seria a correspondente ao caminho escolhido.

Entretanto, existe uma outra semântica para autômatos finitos não-determinísticos descrita por Menezes (2000), que define que “um Autômato Finito é visto como uma máquina composta por fita, unidade de controle e programa. Pode-se afirmar que um *Autômato Finito Não-determinístico* assume um conjunto de estados alternativos, como se houvesse uma multiplicação da unidade de controle, uma para cada alternativa, processando independentemente, sem compartilhar recursos com as demais. Assim, o

processamento de um caminho não influi no estado, símbolo lido e posição da cabeça dos demais caminhos alternativos”. Essa semântica é muito utilizada em linguagens formais, todavia seria interessante aplicá-la também em animações baseadas em autômatos finitos.

Quando se tem uma animação baseada em *autômato finito não-determinístico*, uma das coisas que se deve ter em mente é a forma como este *não-determinismo* será apresentado na tela. O *não-determinismo* pode ser visto como um processo de abertura de janelas no qual, em cada uma delas, a animação será apresentada por um dos caminhos do autômato, conforme mostra a Figura 1. A máquina faz a leitura da fita e, a partir do estado inicial com o símbolo de entrada *a*, tem-se dois caminhos a percorrer (Caminho 1 e Caminho 2). Utilizando a semântica descrita acima, a máquina percorre os dois caminhos do autômato como se fossem dois autômatos independentes e distintos e, para cada um deles, uma nova janela é aberta. A máquina continua funcionando normalmente e, num outro momento, ela tem mais dois caminhos a percorrer, então duas janelas distintas serão abertas. Novamente tem-se autômatos independentes e assim por diante até o final da leitura da fita.

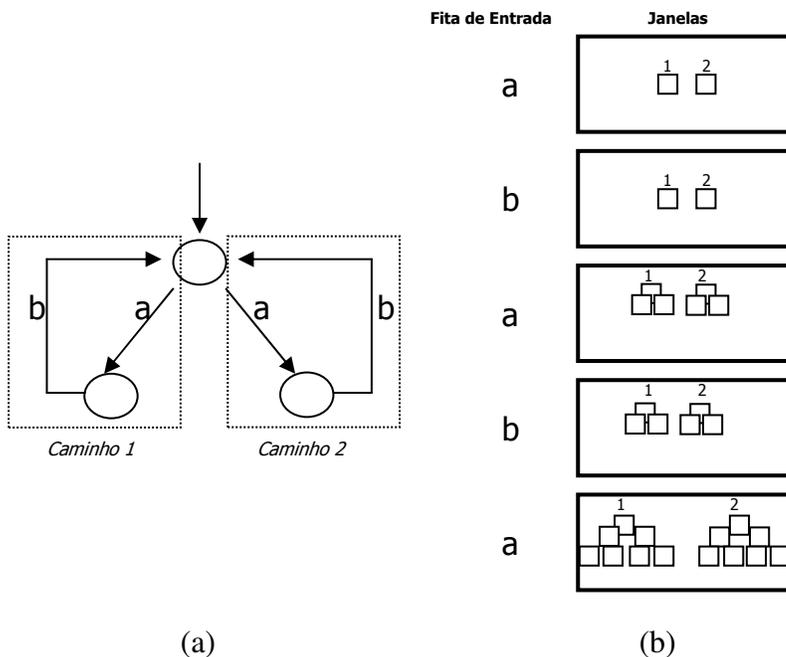


Figura 1 – a) AFND, b) Apresentação do AFND em janelas.

Uma das desvantagens de apresentar as animações dessa forma é que poderá acontecer que, num determinado momento do processamento do autômato, se tenha um laço (finito) e cada uma das janelas abertas acabe abrindo novas janelas sucessivas, podendo ocorrer um estouro delas, e, dependendo da aplicação, poderá não ser interessante o resultado obtido.

Utilizando a mesma semântica da Figura 1, ter-se-iam outras duas formas de apresentação. Numa delas, as animações geradas por cada um dos caminhos do autômato seriam sobrepostas na mesma tela de exibição, o que em algumas aplicações acabaria

dificultando a visualização da animação. Na outra, as animações geradas por caminhos distintos do autômato apareceriam rodando em paralelo na mesma tela de exibição, o que poderia ocasionar um laço (finito) e várias animações em paralelo apareceriam na mesma tela, podendo dessa forma não ser obtido o resultado desejado pelo animador.

Autômato Finito Não-determinístico com pesos (AFNDp)

O autômato utilizado como base para definir o *Autômato Finito Não-determinístico com pesos (AFNDp)* é uma extensão dos autômatos finitos não-determinísticos e a sua definição pode ser encontrada em Aho et al. (1995), Menezes (2000), Hopcroft & Ullman (1969).

O autômato é representado em (MENEZES 2000) como uma 5-upla: $M=(\Sigma, Q, \delta, q_0, F)$, onde Σ corresponde ao alfabeto de símbolos de entrada, Q o conjunto de estados possíveis do autômato, o qual é finito, δ a função programa ou função de transição (função parcial $\delta: Q \times \Sigma \rightarrow 2^Q$), q_0 o estado inicial, tal que q_0 é elemento de Q , e F é o conjunto de estados finais, tal que F está contido em Q .

A extensão proposta no *AFNDp* diz respeito à função programa ou função de transição em que os arcos (transições) do autômato assumem pesos. Esses pesos são definidos pelo projetista da máquina.

A função de transição é denotada como $\delta^P(q, \alpha)$ e a sua definição estendida para $\delta^P: Q \times \Sigma \rightarrow 2^{(Q \times P)}$, em que P é o conjunto de valores dos pesos correspondentes a cada transição. P é o conjunto dos números Naturais menos o zero ($P = \mathbb{N}^{\{0\}}$).

O novo autômato ficaria:

$$M_p = (\Sigma, Q, \delta^P, q_0, F).$$

A Figura 2 representa um *AFNDp* em que cada transição do autômato assume um peso definido pelo projetista da máquina. Se no diagrama os pesos não forem definidos, por convenção é assumido o peso como 1.

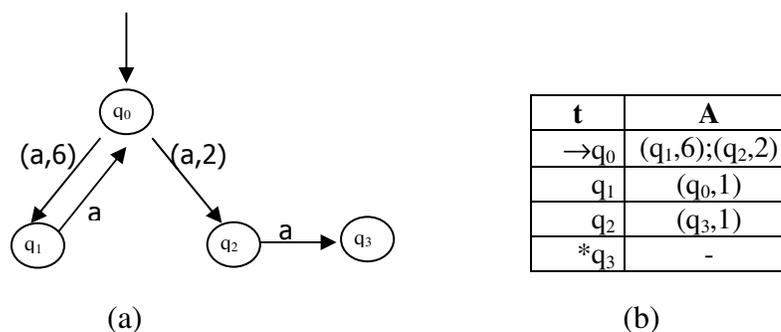


Figura 2 – AFNDp (Autômato Finito Não-determinístico com pesos)
(a) Forma Gráfica (diagrama) e (b) Forma Tabular

No exemplo acima, Figura 2, no estado q_0 com o símbolo de entrada a tem-se duas possibilidades de transição, a máquina escolhe um dos caminhos a seguir, conforme o peso de suas transições (WINSKEL et al., 1996). Será feito um cálculo de cada peso dividido pela soma dos pesos das transições do estado q_0 para o mesmo símbolo de entrada, neste caso a . Tem-se assim $6/8$ e $2/8$, o que é interpretado como sendo a probabilidade da

máquina escolher em 75% dos casos a aresta com peso 6 e em 25% dos casos a outra aresta.

Uma aplicação para o *AFNDp* seria um sistema em que um mesmo rótulo faz referência a diferentes implementações. Pode-se tomar como exemplo uma rede de computadores em que fosse necessário mapear as impressoras em um determinado computador utilizando o *AFNDp*. Poder-se-ia formalizar esta rede conforme ilustra a Figura 3. O cálculo efetuado para os pesos das transições seria o mesmo que foi apresentado anteriormente. Na maioria das vezes o micro envia a impressão para a impressora que tem o peso maior em sua transição.

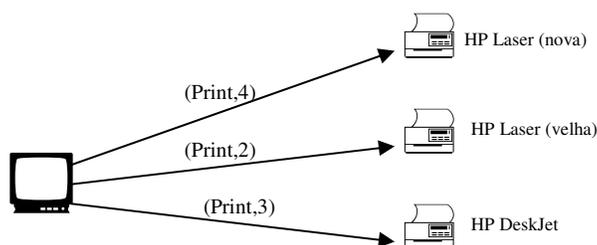


Figura 3 – AFNDp aplicado no mapeamento de impressoras

AFNDp aplicado a animações baseadas em autômatos finitos

O modelo AGA (animação gráfica baseada em autômatos) estudado por Accorsi & Menezes (2000) será utilizado como base para aplicação do *AFNDp*, resultando no AGA-Np (animação gráfica baseada em autômatos finitos não-determinísticos com pesos). O AGA especifica a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação. As especificações em AGA, entretanto, são suportadas por um modelo formal baseado em *autômatos finitos*. Quando os autômatos são simulados mediante a leitura da fita de entrada, as transições entre seus estados controlam a animação dos atores.

O conceito básico de *autômato finito* limita a saída a uma informação binária: aceita ou rejeita. Entretanto, há duas abordagens: Máquina de Mealy e Máquina de Moore (HOPCROFT et al., 1979, MEMEZES, 2000), que estendem esse conceito para modelos que possibilitam a geração de palavras como saída. A Máquina de Mealy associa as saídas às transições, enquanto a Máquina de Moore os associa aos estados.

Devido à equivalência entre as duas abordagens, o ator AGA pode ser estendido tanto vinculando a saída às transições quanto aos estados. Nos artigos publicados por Accorsi, as duas abordagens são exploradas. Em Accorsi & Menezes (2000), o modelo AGA é apresentado utilizando como base a Máquina de Moore e, em Accorsi et al. (2001), utilizando a abordagem de saída proposta pela Máquina de Mealy combinada ao modelo de autômato temporizado.

O *AFNDp* poderá ser aplicado nas duas abordagens descritas, contudo foram preferidas para efeito de ilustração as animações vinculadas a máquina de Moore. A Figura 4 ilustra a estrutura básica do modelo AGA aplicada a uma animação com três atores. Os atores *mar*, *sol* e *ave* são especificados a partir de *autômatos finitos* em que as imagens

estão associadas aos estados. Essas imagens correspondem a variações gráficas que o ator pode sofrer durante a animação.

A área de atuação do autômato é dividida em camadas com o objetivo de estabelecer a ordem de sobreposição das imagens de saída. Assim, a imagem correspondente é formada pela unificação das camadas.

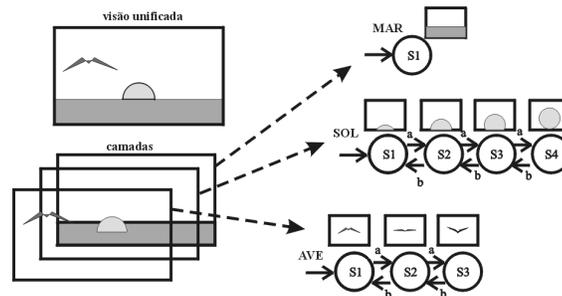


Figura 4 – Modelo de animações do AGA (Accorsi 2000).

No AGA, todas as animações projetadas são baseadas em *autômatos finitos determinísticos* e o poder de expressão das animações acaba sendo limitado pela sua utilização. Na maior parte dos casos, é mais simples expressar uma animação de forma *não-determinística* do que de forma *determinística*.

Pode-se tomar como exemplo o caso em que o projetista deseja criar um boneco caminhando e que na maioria das vezes inicia sua caminhada com a perna direita e outras vezes com a perna esquerda. O autômato que representa o movimento da perna do boneco poderá ser montado a partir de um *AFND*, conforme ilustra a Figura 5. O autômato finito determinístico equivalente para este caso não conseguirá expressar o objetivo do projetista. Uma solução seria montar dois autômatos finitos determinísticos distintos, um com a caminhada do boneco iniciando com a perna direita e outro com a caminhada do boneco iniciando com a perna esquerda, e alternar a visualização de cada um dos autômatos. O objetivo do projetista seria alcançado, porém, ter-se-ia dois autômatos ao invés de um único que solucionasse o caso.

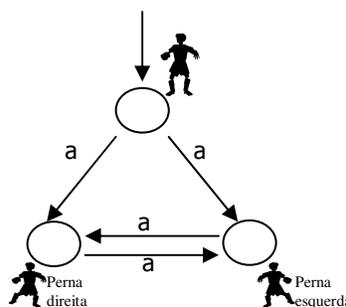


Figura 5 – Representação do movimento de um boneco caminhando em um AFND

A animação apresentada na Figura 5 consegue representar o movimento do boneco de forma clara e simplificada, contudo as semânticas de *AFND* já apresentadas, ainda assim,

não possibilitam obter o resultado desejado pelo projetista da animação. Utilizando o *AFNDp*, o resultado poderá ser alcançado. Neste novo autômato, o projetista deverá definir pesos para cada uma das arestas. Sendo assim, o maior peso seria dado a aresta que fizesse o movimento da perna direita e um peso menor seria atribuído a aresta que representa o movimento da perna esquerda, conforme ilustra a Figura 6.

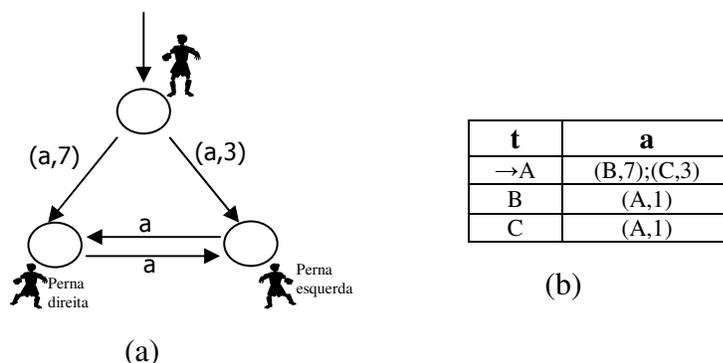


Figura 6 – Representação de um boneco caminhando usando o AFNDp: a) forma Gráfica e b) forma Tabular.

Neste caso, o *AFNDp* calcularia o valor dos pesos conforme explicado anteriormente e ter-se-ia em 70% das vezes o autômato percorrendo a aresta com peso 7 e 30% a aresta com peso 3, isto significa que a máquina escolheria em 70% das vezes que o boneco se movimentasse com a perna direita e 30% com a perna esquerda.

Conclusões

Uma das vantagens de se utilizar *autômatos finitos não-determinísticos* é que esse pode ser muito menor em número de estados e transições do que o seu *autômato finito determinístico* equivalente (AHO et al., 1995). Entretanto, as semânticas dos *autômatos finitos não-determinísticos*, apresentadas neste artigo, em muitos casos não conseguem obter os resultados esperados pelo projetista da máquina, principalmente em animações computacionais.

O modelo proposto, *AFNDp* (Autômato Finito Não-determinístico com pesos), aumenta o poder de expressão dos *autômatos finitos não-determinísticos* utilizando a noção de pesos em suas arestas. Na semântica proposta, a máquina escolhe aleatoriamente o caminho a seguir a partir dos pesos especificados pelo projetista da mesma, que é responsável pela definição da prioridade de cada um dos caminhos. Dessa forma, os resultados, muitas vezes não obtidos com a aplicação dos *autômatos finitos não-determinísticos*, são obtidos utilizando o *AFNDp*.

A aplicação do *AFNDp* no modelo AGA tem como objetivo estender o modelo para que se possam criar animações que serão expressas de forma simples (menos estados e menos transições), o que na maioria das vezes não se consegue utilizando o *autômato finito determinístico* equivalente. O modelo originado pela aplicação do *AFNDp* no AGA resulta no AGA-Np (animação gráfica baseada autômato finito não-determinístico com pesos).

O modelo AGA utilizado para efeito de interpretação do *AFNDp* é baseado na Máquina de Moore (ACCORSI & MENEZES, 2000), porém a sua aplicação em animações baseadas na Máquina Mealy (ACCORSI et al., 2001) poderá ser feita sem nenhum problema, pois os dois modelos são equivalentes.

Muitas extensões podem ser feitas no modelo AGA, e a implementação de um protótipo do AGA-Np já está sendo efetuada. Está em estudo a semântica de algumas operações categoriais sobre o modelo AGA, de forma que essas operações podem vir a ser aplicadas no AGA-Np.

Referências Bibliográficas

AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey. *Compiladores. Princípios, Técnicas e Ferramentas*. Trad. Daniel de Ariosto Pinto. Rio de Janeiro: LTC, 1995.

ACCORSI, Fernando e MENEZES, Paulo Blauth. *Animação Gráfica Baseada na Teoria de Autômatos*. In: WMF – WORKSHOP ON FORMAL METHODS, SBC, 2000, João Pessoa, Paraíba. Proceedings... p. 122-127.

ACCORSI, Fernando; MENEZES, Paulo Blauth; NEDEL, Luciana Porcher. *Animação Gráfica Baseada em Autômatos Temporizados Sincronizados*. In: WMF - WORKSHOP ON FORMAL METHODS, 2001, Rio de Janeiro. Proceedings...

DROBOT, V. *Formal Languages and Automata Theory*. New York: Computer Science Press, 1989.

GERSTING, Judith L. *Fundamentos Matemáticos para a Ciência da Computação*. 3. ed. São Paulo: LTC, 1995.

HOPCROFT, J. E.; ULLMAN, J. D. *Formal Languages and Their Relation to Automata*. Reading: Addison Wesley, 1969.

HOPCROFT, J. E. ; MOTWANI, R. ; ULLMAN, J. D. *Introduction to Automata Theory, Languages and Computation*. Reading: Addison Wesley, 1979.

MENEZES, Paulo Blauth Menezes. *Linguagens Formais e Autômatos*. 3. ed. Instituto de Informática da UFRGS. Porto Alegre: Sagra Luzzatto, 2000.

WINSKEL, Glynn; SASSONE, Vladimiro; NIELSEN, Mogens. *Models for concurrency: towards a classification*. Theoretical Computer Science, v. 170, n. 1-2, p. 297-348, 1996.

ANEXO C PROJETO PNEE - AUTÔMATO

Conselho Nacional de Desenvolvimento Científico e Tecnológico
- CNPq

Dados do Projeto e do Proponente

Sigla:	PNEE – Autômato
Título do Projeto:	PNEE - Autômato: Sistema Hipermídia para Portadores de Necessidades de Educação Especial, baseado em Teoria dos Autômatos
Referência da Chamada:	Chamada CT-INFO: CNPq 2003 – SOFTWARE LIVRE Programa de Pesquisa e Desenvolvimento Tecnológico em Software Livre
Linha(s) de atuação em que se insere o projeto/Grupo:	<input checked="" type="checkbox"/> Pesquisa <input checked="" type="checkbox"/> Desenvolvimento de aplicações <input type="checkbox"/> Projetos de demonstração <input checked="" type="checkbox"/> Inovação tecnológica <input checked="" type="checkbox"/> Transferência de tecnologia <input checked="" type="checkbox"/> Capacitação tecnológica <input type="checkbox"/> Estudos prospectivos e outros
Coordenador do Projeto:	Paulo Fernando Blauth Menezes
Instituição Executora:	UFRGS
Data:	29/10/2003

1. Identificação e caracterização do problema

A Internet como um meio de ensino tem proporcionado o desenvolvimento de vários *softwares* educacionais com diferentes propostas pedagógicas e tecnologias de informação e comunicação [2] [3] [10] [11] [13] [14] [15] [16]. Os pesquisadores buscam a melhoria e eficácia de *softwares* já existentes, bem como a construção de novos *softwares*. Estes *softwares*, desenvolvidos para diversos fins e objetivos, abrangendo pesquisadores de áreas tais como pedagogia, psicologia, terapia ocupacional, informática entre outras, têm sido objeto de inúmeras pesquisas na atualidade. Em particular, e mais recentemente, tem sido dada especial atenção à Educação Especial.

Com a expansão da Internet e o crescente desenvolvimento de tecnologias, tanto de hardware quanto de software, para a *web*, faz com que um grande número de pessoas com necessidades distintas tenham acesso a informações e também utilizem a internet como um meio de ensino-aprendizagem. Devido à diversidade de pessoas que utilizam a internet como meio de ensino, existem várias pesquisas que buscam cada vez mais desenvolver tecnologias, metodologias e ferramentas que atendam as necessidades destas pessoas, que podem ser pessoas com necessidades especiais, déficits cognitivos e com diferentes preferências cognitivas.

Levando-se em conta o número significativo de portadores de necessidades especiais, que segundo o Censo de 2000 do Instituto Brasileiro de Geografia e Estatística - IBGE, são 5.685.956 portadores de necessidades especiais - PNE, com dificuldades de audição no Brasil. Entre esta população encontram-se os surdos oralizados (que não fazem uso das Línguas de Sinais) e os surdos não oralizados (que se utilizam da Língua Brasileira de Sinais – LIBRAS, entre outras, para se comunicarem). Em 2002 o Congresso Nacional e a Presidência da República aprovaram e sancionaram a Lei n.º 10.436, de 24 de abril de 2002, que reconhece a LIBRAS como meio legal de comunicação e expressão da comunidade surda.

O contexto apresentado acima, experiências anteriores de pesquisadores inseridos neste projeto e o sentimento de contribuição com a sociedade, inserindo cidadãos especiais no mundo digital constituem a motivação social do presente trabalho. Do ponto de vista técnico, a ferramenta aqui proposta significa uma união de esforços de grupos de interesse no tema.

Observe-se, também, que foi constatado que uma grande maioria dos softwares para desenvolver aplicações para portadores de necessidades especiais possuem limitações que influenciam significativamente no aprendizado de deficientes auditivos. Assim, algumas dessas limitações, identificadas na literatura serão supridas no projeto proposto:

- Hiper-animações. Uma hiper-animação não se limita às animações tradicionais: combina imagens, vídeos digitais, animações gráficas e sons com dispositivos de entrada não usuais a fim de realizar um alto nível de interatividade [KAP 91, p. 101].
- Software livre, possibilitando a distribuição do produto para a comunidade.
- web (independência de plataforma, tecnologia, etc)

Buscou-se o desenvolvimento de uma ferramenta que permita criar aplicações voltadas à educação especial, especificamente, deficientes auditivos e que contemplem as funcionalidades necessárias para o público alvo. Destaque-se as facilidades de hiper-animações o que constituiriam um requisito ideal para desenvolver aplicações para este tipo

de deficiente. Esta ferramenta possui construções especialmente projetadas para o desenvolvimento de aplicações que tratem:

- ensino/aprendizagem de deficientes auditivos
- ensino /aprendizagem de deficientes auditivos x ouvintes
- comunicação entre deficientes auditivos
- comunicação entre deficientes auditivos e ouvintes

Portanto, trata-se de uma importante ferramenta para a integração de Portadores de Necessidades Educação Especial (PNEE) na sociedade. É importante ressaltar que o uso de tal ferramenta não se limita ao desenvolvimento de aplicações para PNEE. Por exemplo, aplicações aos moldes de taquígrafos, seriam perfeitamente cabíveis.

A ferramenta a ser desenvolvida utilizará um modelo hipermídia baseado na representação da estrutura de uma hiperbase através de formalismos da Teoria de Autômatos[10].Este modelo baseia-se no fato de que autômatos não somente capturam o poder descritivo dos grafos diretos, conhecidos por serem uma abstração útil em sistemas de hipertexto [5], como também fornecem uma máquina abstrata para o controle e análise de hipertextos e animações, além de ser um formalismo universalmente aceito e possuir uma história de uso em muitas disciplinas como uma técnica de diagramação formal [7, 12]. Os autômatos podem ser representados por uma estrutura de grafo, a qual pode ser manipulada sobre vários aspectos. Grafos com uma semântica formal podem ser usados para prover interfaces de programação para o controle de material hipermídia. Diversos trabalhos relacionados ao uso de representações gráficas na modelagem de aplicações hipermídia utilizam outras construções, como Mapas Conceituais [8], Workflow [2], modelos Entidade-Relacionamento, Orientação a Objetos [9], e também Redes de Petri (o modelo Trellis [7] é um exemplo). Essa ferramenta utilizará também tecnologias dentro da categoria de software livre para que o protótipo desenvolvido possa ser distribuído tanto para comunidade acadêmica quanto para a sociedade.

É importante observar que o uso da Teoria dos Autômatos não limita a ferramenta em questão apenas para desenvolvimento de aplicações para deficientes auditivos. De fato, com uma simples substituição de alfabeto de entrada (operações triviais neste modelo), é possível desenvolver aplicações para, por exemplo, portadores de paralisia cerebral.

Adicionalmente, busca-se fazer uma revisão bibliográfica sobre as técnicas utilizadas no desenvolvimento de interfaces, preocupando-se com conceitos como: ergonomia, usabilidade, acessibilidade, padrões de projeto, etc., com o objetivo de contemplar características que envolvem a construção de um software educacional para educadores desenvolverem aplicações para deficientes auditivos.

2. Justificativa

A pesquisa científica permite que sejam desenvolvidos trabalhos que contribuam para o desenvolvimento social e acadêmico de uma nação, o que é muito importante, pois proporciona uma disseminação dos conhecimentos descobertos. Dentro deste contexto, o ensino-aprendizagem via www gerou um grande interesse da comunidade científica. No entanto, o crescente desenvolvimento de tecnologias permite que cada vez mais pessoas tenham acesso ao ensino a distância (EAD), porém muitos ambientes voltados ao ensino não apresentam as propriedades adequadas ao aprendizado, principalmente, ambientes voltados à educação especial.

Como o EAD independe de região geográfica, pois as pessoas podem acessar a internet de qualquer local, sendo por comodidade ou por necessidade, o conhecimento acaba atingindo um número maior de pessoas. Devido a esse grande número de pessoas e à individualidade de cada uma, existem também diversas pesquisas tanto ao nível de desenvolvimento de softwares voltados à educação quanto à educação especial. Porém, na educação especial, existem ainda poucos grupos de pesquisas trabalhando com o desenvolvimento de software [4], o que provoca uma carência nesta área, levando-se em conta ainda que a maioria dos softwares não são projetados com os requisitos básicos para atender aos portadores de deficiências.

Com isso, a proposta de desenvolver uma ferramenta baseada em fundamentações consolidadas faz com que o projeto invista em pesquisas, que busquem aperfeiçoar os módulos já desenvolvidos em um ambiente web voltado ao desenvolvimento de aplicações para portadores de deficiência auditiva, podendo ser estendido para portadores de necessidades especiais como: paralisia cerebral, visual.

Esta ferramenta será desenvolvida utilizando tecnologias de software livre e será voltada para a web, tendo como preocupação contemplar as diversas peculiaridades de um software voltado à educação especial. Tendo como uma das preocupações desenvolver uma interface que permita o usuário executar, igualmente, qualquer função do Sistema através do teclado ou através do dispositivo apontador, melhorando a ergonomia do software. Com isso, obteremos um ambiente gráfico, multimídia, dinâmico e interativo para Internet baseado em hiper-animações e autômatos finitos com saída, tendo como finalidade básica permitir que educadores desenvolvam aplicações para portadores de necessidades especiais auditivas via Internet.

Como resultado, espera-se que este ambiente contribua para o avanço do estado-da-arte em desenvolvimento e formalização de sistemas hipermídia e induza novas ferramentas e métodos voltados a educação especial, permitindo o desenvolvimento de aplicações adequadas aos portadores de deficiência e possibilitando que os educadores possam desenvolver aplicações de boa qualidade, permitindo uma real integração do PNEE à sociedade.

3. Objetivos e Metas

a) Objetivos

Baseado nos modelos formais desenvolvidos e nos resultados atingidos em três projetos de pesquisa desenvolvidos pelo Grupo, a saber:

- Hyper-Automaton, um sistema adaptativo de disponibilização de conteúdo hipertexto;
- AGA, Animação 2D Baseada em Autômatos para a Web;
- Astrha, um Ambiente Gráfico, Dinâmico e Interativo para Internet Baseado em Hiper-Animações e na Teoria dos Autômatos;

Os principais objetivos são os seguintes:

- 1) Desenvolver uma ferramenta voltada ao desenvolvimento de aplicações para portadores de deficiência auditiva, podendo ser estendido para portadores de necessidades especiais como: portadores de Deficiência Mental, Deficiência Visual, Deficiência Motora.

Esta ferramenta será baseada em tecnologias de software livre e será voltada para a web, tendo como uma das preocupações uma interface que permita o usuário executar, igualmente, qualquer função do Sistema através do teclado ou através do dispositivo apontador, melhorando a ergonomia do software. Com isso, obteremos um ambiente gráfico multimídia, dinâmico e interativo para Internet baseado em hiper-animações e autômatos finitos com saída, tendo como finalidade básica permitir que educadores desenvolvam aplicações para portadores de necessidades especiais auditivas via Internet;

2) Validar esta ferramenta num contexto típico de instituição de ensino, junto ao Instituto de Informática da UFRGS.

b) Metas

SEMESTRE 1: Revisão bibliográfica, especificações e implementação da ferramenta.

SEMESTRE 2: Implantação e validação com estudos de casos.

4. Relevância dos resultados e impactos esperados

a) Indicadores de resultados (ao final do projeto)

Pretende-se ao final do projeto ter produzido e publicado um bom número de artigos científicos que apontem para a colaboração dos resultados teóricos na comunidade científica internacional e nacional. Artigos descrevendo a aplicabilidade do protótipo de ferramenta a ser construído e os estudos de caso validarão a sua adequação como proposta para seu posterior desenvolvimento nas instituições de Ensino Especial.

O desenvolvimento deste projeto, principalmente na área teórico-formal, será fortemente centrado na formação de doutores, mestres e graduados em Ciência da Computação com forte apoio de terceiros. Como se trata de um projeto que em diversos momentos é multidisciplinar, será dada continuidade à política correntemente utilizada pelo Grupo de colaboração e/ou co-orientação com outros grupos complementares, tanto da UFRGS quanto de outras instituições.

O produto da aplicação chave constitui-se em uma ferramenta que utilizará um modelo hipermídia baseado na representação da estrutura de uma hiperbase através de formalismos da Teoria de Autômatos. Busca-se o desenvolvimento de uma ferramenta que permita criar aplicações voltadas à educação especial, especificamente, deficientes auditivos e que contemplem as funcionalidades necessárias para o público alvo. Destaque-se as facilidades de hiper-animações o que constituiriam um requisito ideal para desenvolver aplicações para este tipo de deficiente.

b) Impactos Esperados

Como resultado, espera-se que este ambiente contribua para o avanço do estado-da-arte em desenvolvimento e formalização de sistemas hipermídia e induza novas ferramentas e métodos voltados à educação especial, permitindo o desenvolvimento de aplicações adequadas aos portadores de deficiência e possibilitando que os educadores possam desenvolver aplicações de boa qualidade, permitindo uma real integração do PNEE à sociedade.

5. Considerações finais

O grupo de pesquisa proponente possui experiência na formalização de modelos e desenvolvimento de sistemas para a WWW como demonstram algumas das publicações relevantes a este projeto, obtidas por seus integrantes e listadas no item Referências Bibliográficas (a lista completa das publicações pode ser obtida na Plataforma Lattes do CNPq). Observe-se que diversas publicações são em conjunto com outros grupos de pesquisa de instituições nacionais e internacionais.

O coordenador do projeto possui também forte atuação na formação de recursos humanos nesta área de pesquisa, incluindo a conclusão, no último ano, da orientação de três projetos de graduação e quatro trabalhos de mestrado, além de outros em andamento.

O principal produto já desenvolvido pelo grupo encontra-se na página <http://teia.inf.ufrgs.br> e se constitui em um sistema semi-automatizado para a organização de hiperdocumentos instrucionais e um *player* para animações baseadas em autômatos. Este sistema está atualmente sendo utilizado como forma de suporte ao ensino de disciplinas no curso de Ciência da Computação da UFRGS e pode ser encontrado no link <http://teia.inf.ufrgs.br>.

Finalmente, devemos lembrar o desafio que representa o trabalho proposto neste projeto. A área envolve uma grande quantidade de conhecimentos de áreas distintas, evidenciando-se assim um caráter interdisciplinar e multidisciplinar.

6. Referências Bibliográficas

[1] ACCORSI, F.; MENEZES, P. F. Blauth; NEDEL, L. P. Animação Gráfica Baseada em Autômatos Temporizados Sincronizados. In: WMF'2001: IV Workshop de Métodos Formais, 2001, Rio de Janeiro. Anais do IV Workshop de Métodos Formais. Rio de Janeiro: SBC, 2001. v. 1, p. 75-80.

[2] ADELSBERGER, H.; KÖRNER, F. X.; PAWLOWSKI, J. M. A Conceptual Model for an Integrated Design of Computer Supported Learning Environments and Workflow Management Systems. In: XV IFIP World Computer Congress, 1998, Vienna. Proceedings of the XV IFIP World Computer Congress - Teleteaching 98. Vienna: IFIP, 1998. p.55-64.

[3] CARNEIRO, C. R. J. B.; REIS, R. Q.; MENEZES, P. F. Blauth. Especificação Formal de uma Ferramenta de Trabalho Colaborativo através da Composição de Objetos Nautilus. In: SBES'99: XIII SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 1999, Florianópolis. Anais do XIII Simpósio Brasileiro de Engenharia de Software. Florianópolis: SBC, 1999. p.95-110.

[4] DIVERIO, T. A., MITO, I. V. Foundations for Virtual Enviroments to Support Theory of Computation Teaching In: Advances Educational Technologies: multimedia, www and distance education ed. New York : John Wiley & Sons, 2001

[5] DIVERIO, T. A.; MENEZES, P. F. B. Teoria da Computação: máquinas universais e computabilidade. 2 ed. Porto Alegre : Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2000, v.1. p.224.

[6] FREITAS, C. M. D. S., CHUBACHI, O. M., LUZZARDI, P. R. G., CAVA, R. A. Introdução à Visualização de Informações. Revista de Informática Teórica e Aplicada. Porto Alegre, RS, v.8, n.2, p.143 - 158, 2001.

[7] FURUTA, R.; STOTS, P. D. Programmable Browsing Semantics in Trellis. In: Proceeding of the Hypertext'89, 1989, p.27-42.

[8] GAINES, B. R.; SHAW, M. L. G. Concept Maps as Hypermedia Components. 1996. <http://ksi.cpsc.ucalgary.ca/articles/ConceptMaps/>

[9] HAEUSLER, E. H.; MESEGUER, J. Mathematical and Engineering Foundations for Interoperability through Architecture. In: Proceedings of the Projects Evaluation Workshop CNPq/NSF/Inria, Ed. Nelson Prugner, Paulo Ernesto Castilho Lima e Celso Deusdeti Costa., p. 175-208, 2001.

- [10] HOPCROFT, J. E.; ULLMAN, J. D., Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.
- [11] MACHADO, J. H. A. P.; MORAIS, C. T. Q. de; MENEZES, P. F. Blauth; REIS, R. A. da Luz. Structuring Web Course Pages as Automata: revising concepts. In: RIAO'2000: Recherche d'Informations Assistée par Ordinateur, 2000, Paris. Content-Based Multimedia Information Access: conference proceedings. Paris: C.I.D., C.A.S.I.S., 2000. v.1, p. 150-159.
- [12] MAURER, H. Necessary Ingredients of Integrated Network Based Learning Environments. In: ED-MEDIA/ED-TELECOM Educational Multimedia/Hypermedia and Telecommunications, Charlottesville, v.1, 1997, p.619-624.
- [13] MENEZES, P. F. B., SERNADAS, A., COSTA, J.F., Nonsequential Automata Semantics for Concurrent, Object-Based Language. In First US-BRAZIL Workshop on Formal Foundations of Software Systems, 1998, Rio de Janeiro Eletronc Notes in Theoretical Computer Science, Amsterdan Elsevier, 1998, v. 14.
- [14] MENEZES, P. F. B.; COSTA, J. F. Synchronization in Petri Nets, Fundamenta Informaticae, Amsterdan, v. 26, n.1, p. 11-22, 1996.
- [15] MORAIS, C. T. Q. de; MACHADO, J. H. A. P.; MENEZES, P. F. B.; REIS, R. A. da Luz. A Web Teaching System Based on Formal Methods. In: WCC'2000: 16th IFIP World Computer Congress, 2000, Beijing. Proceedings ICEUT, IFIP World Computer Congress. Beijing: PHEI, 2000. v. 1, p. 221-224.
- [16] NEDEL, L. P.; THALMANN, D. Anatomical modeling of deformable human bodies. The Visual Computer, Heidelber, Germany, v.16, n.6, p.306-321, 2000.