

# Um Levantamento da Evolução da Adaptatividade e da Tecnologia Adaptativa

J. J. Neto

**Resumo** — Esta publicação procura analisar a evolução da adaptatividade e de suas aplicações. Neste contexto, o termo se refere à propriedade que um sistema ou um dispositivo apresenta de modificar suas próprias regras de funcionamento, em função de seu histórico, sem interferência externa. Entende-se como tecnologia adaptativa a aplicação da adaptatividade com fins práticos. Este artigo levanta uma parcela significativa da literatura publicada em torno da adaptatividade e de assuntos associados e procura relacionar as obras segundo os temas, e cada tema, segundo a cronologia de publicação.

**Palavras-chave** — Adaptatividade, Dispositivos Adaptativos, Sistemas Adaptativos, Sistemas Automodificáveis.

## I. INTRODUÇÃO

ADAPTATIVIDADE, conforme a conotação aqui utilizada, é um termo que se refere a um conceito bastante simples: a capacidade que tem um sistema de, sem a interferência de qualquer agente externo, tomar a decisão de modificar seu próprio comportamento, em resposta ao seu histórico de operação e aos dados de entrada. Assim, a “experiência anterior” adquirida por um sistema ou dispositivo adaptativo é decisiva quanto ao tipo de alteração comportamental resultante do exercício da adaptatividade, e duas instâncias idênticas de um mesmo sistema adaptativo podem evoluir para comportamentos finais completamente diferentes, de acordo com a diversidade dos eventos a que forem submetidas em suas operações.

O termo *dispositivo*, a que se refere o texto deste artigo, corresponde a alguma abstração formal, na qual o comportamento do dispositivo é regido por um conjunto finito explícito de regras que especificam, para cada situação em que se encontra o dispositivo, sua nova situação e também as correspondentes alterações esperadas no conjunto de regras que o definem.

O restante do artigo está organizado da seguinte forma: em uma seção, citam-se os principais conceitos tratados; em outra, apresentam-se sete subseções, classificando-se os artigos segundo os tipos de dispositivos em que se baseiam; na seguinte, doze subseções apresentam os principais tipos de aplicações da tecnologia adaptativa; finalmente, uma seção relaciona

outros recursos sobre adaptatividade, seguindo-se uma conclusão.

## II. CONCEITOS PRINCIPAIS

Na construção de dispositivos formais adaptativos dirigidos por regras, conceitualmente é possível [25] separar de forma clara duas componentes importantes: um dispositivo subjacente, tipicamente não-adaptativo, e um mecanismo adaptativo, responsável pela incorporação da adaptatividade.

Na seção III deste artigo, procurou-se agrupar as referências bibliográficas aqui relacionadas de acordo com os formalismos subjacentes correspondentes aos dispositivos adaptativos a que se referem os artigos.

Os formalismos subjacentes dos dispositivos adaptativos podem ser classificados em variadas categorias, conforme sua forma de operação. Entre outras, têm-se as seguintes:

- *dispositivos de reconhecimento*, da classe dos autômatos (seção *E*), baseados na sucessão de mudanças de estados;
- *dispositivos de geração*, da classe das gramáticas (seção *F*), baseados na aplicação sucessiva de regras de substituição;
- *dispositivos para a representação de sistemas assíncronos*, tais como os *statecharts* (seção *G*), que incorporam mecanismos responsáveis pela representação de fenômenos de sincronização (seção *G*);
- *dispositivos estocásticos*, como as redes de Markov (seção *G*), capazes de representar fenômenos de caráter aleatório;
- *dispositivos de auxílio à tomada de decisões*, representados principalmente pelas tabelas de decisão e pelas árvores de decisão (seção *G*);
- *dispositivos de processamento*, como as linguagens de programação adaptativas, que permitem descrever a lógica de programas com código automodificável.

Na seção IV, é possível visualizar com maiores detalhes uma série de aplicações dos formalismos adaptativos associados a essas diversas classes de formalismos subjacentes, incluindo: inferência, arte usando computador, processamento de linguagem natural, síntese de voz, reconhecimento de padrões, tomada de decisão, linguagens de programação adaptativas, otimização de código, metamodelagem, computação evolutiva, engenharia de software e robótica.

## III. REVISÃO DA LITERATURA

Esta seção apresenta um panorama, em nível internacional, de uma significativa parcela da literatura disponível acerca das diversas manifestações da adaptatividade.

Original recebido em 31 de agosto de 2007.

João José Neto é o responsável pelo LTA – Laboratório de Linguagens e Tecnologia Adaptativa, vinculado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo – Av. Prof. Luciano Gualberto, trav. 3, No. 158 – Cidade Universitária – São Paulo - SP - Brasil. (joao.jose@poli.usp.br) fone: (11) 3091-5402

Alguns tópicos não diretamente relacionados com a adaptabilidade têm ou tiveram papel marcante na evolução desse conceito e aqui foram incluídos por sua importância histórica nesse processo, como é o caso dos autômatos de pilha estruturados e da extensibilidade.

A coleta de referências aqui apresentada se apóia amplamente naquelas que resultaram de diversos esforços anteriores, publicados por autores tais como Iwai [32], Pistori [33], Pariente [38], Shutt [45] e Jackson [49].

#### A. Autômatos de pilha estruturados

Antes de iniciar a resenha de publicações sobre a adaptabilidade, convém estudar os autômatos de pilha estruturados, um formalismo que, embora não tenha características adaptativas, exerceu um importante papel no desenvolvimento dos trabalhos sobre a adaptabilidade na Escola Politécnica.

Em 1963, Conway publicava um famoso trabalho [74], que, muito mais tarde, viria a inspirar a concepção primeiramente dos autômatos de pilha estruturados [27] e depois, dos autômatos adaptativos, deles derivados [2], [3], [43].

Nesse artigo, Conway apresentou os *diagramas separáveis de transições*, que consistem essencialmente de um conjunto de autômatos finitos mutuamente recursivos, que assim interligados têm poder computacional suficiente para o tratamento de linguagens livres de contexto.

O trabalho de Conway [74], de grande importância prática, carecia, no entanto, de uma fundamentação teórica explícita, o que motivou Lomet a publicar mais tarde seu artigo formalizando os diagramas separáveis de transições [75].

Em 1981, no Brasil, Neto e Magalhães publicavam o primeiro artigo [27] sobre o formalismo dos *autômatos de pilha estruturados*, que viriam a se tornar a base dos autômatos adaptativos, cuja idéia foi esboçada mais tarde [43], em 1988.

O uso principal dos autômatos de pilha estruturados seria, no entanto, o formalismo de base para o funcionamento de reconhecedores sintáticos da componente livre de contexto de linguagens de programação. Esse trabalho constituiu o alicerce de uma frutuosa prática de ensino da construção de compiladores dirigidos por sintaxe, cujos reconhecedores sintáticos subjacentes são autômatos de pilha estruturados.

Os autômatos de pilha estruturados são dispositivos reconhecedores constituídos de uma família de submáquinas, cujas transições podem ser internas, como as de um autômato finito, ou então responsáveis pela movimentação entre submáquinas (chamadas e retornos). Neste caso, uma pilha é utilizada, estritamente para memorizar estados de retorno, os quais são empilhados sempre que uma submáquina é chamada, para memorizar o estado para o qual ela deve retornar ao final da operação da submáquina assim acionada. Ao final da operação da submáquina, tal estado de retorno é desempilhado e utilizado para promover o prosseguimento da operação interrompida da submáquina chamadora.

O estudo dos autômatos de pilha estruturados motivou Neto a publicar, em 1987, um livro introdutório sobre compiladores [26], no qual descreve um método prático e eficiente de obtenção, a partir de gramáticas livres de contexto denotadas na

Notação de Wirth, de autômatos de pilha estruturados, para uso como núcleos de compiladores dirigidos por sintaxe.

Em 1999, Neto, Pariente e Leonardi apresentam um artigo [28] que descreve cuidadosamente um processo de automatização desse método, indicando o caminho para a construção de um gerador automático de reconhecedores sintáticos para linguagens livres de contexto.

Mais tarde, em 2005, Ricchetti publicaria sua dissertação [81], na qual esse processo de automatização é estendido, de forma que seja gerada também a árvore de *parsing* do texto em análise, montada de acordo com a gramática da linguagem.

A evolução posterior dos autômatos de pilha estruturados lhe deu a potência de representação suficiente para tratar linguagens dependentes de contexto, levando à concepção dos autômatos adaptativos, esboçados em 1988 [43].

Em 1999, Pereira apresenta sua dissertação de mestrado [34], na qual introduz uma notação gramatical com o poder de descrição de linguagens dependentes de contexto, mas que pode ser utilizada também para linguagens livres de contexto. Com essa última conotação, Pereira ilustra sua proposta descrevendo um *bootstrap* da própria ferramenta apresentada.

#### B. Dependências de contexto

A mais antiga menção, na literatura, a processos adaptativos é provavelmente a encontrada no trabalho que Di Fiorino publicou em 1963, no qual relata a idéia da utilização de um conjunto dinâmico de regras para a definição sintática de linguagens [63]. Sua gramática, baseada em notações livres de contexto, é composta de produções ditas gerais, que formam um conjunto fixo, e de produções locais, que são construídas conforme a necessidade particular da sentença analisada.

O tema da dependência de contexto, encarada como sintaxe, levou a muitas discussões na literatura. Em 1980, McGettrick já apresentava em seu livro [40], sobre a definição de linguagens de programação, uma distinção entre semântica estática e semântica dinâmica, esclarecendo que a dita semântica estática, assim como a sintaxe, corresponderia a uma atividade desenvolvida pelo compilador, enquanto a semântica dinâmica seria responsável pelas ocorrências durante a execução dos programas.

Em 1981, Pagan, em seu livro sobre especificações formais de linguagens de programação [37], faz uma comparação de diversos formalismos conhecidos à época e, em particular, levanta novamente a discussão sobre a inadequação do termo *semântica estática* para designar *dependências de contexto*.

Em seu artigo de 1990, Meek [72] argumenta fortemente contra o uso desse termo e o aponta como evidência de uma forma incorreta de encarar um fenômeno genuinamente sintático, que é a dependência de contexto. Slonneger e Kurtz, em seu livro de 1995 sobre sintaxe e semântica das linguagens de programação [41], tecem também considerações que reforçam essa posição. Em seu artigo de 2003, sobre o uso de autômatos em engenharia de computação [42], Neto também tece considerações a esse respeito, defendendo esse mesmo princípio e, para isso, citando argumentos encontrados em Pagan [37], Slonneger [41] e McGettrick [40].

Está cada vez mais disseminada a idéia de incluir nos formalismos que definem as linguagens dependentes de contexto informações mais completas que aquelas usualmente representadas apenas por sua componente livre de contexto. Assim sendo, a tendência é de que o tratamento das dependências de contexto seja cada vez menos atribuído a atividades semânticas, e que esse aspecto lingüístico seja reconhecido cada vez mais como sintaxe autêntica, reservando-se para a semântica aquilo que lhe é mais próprio, ou seja, os aspectos que se referem estritamente à interpretação da linguagem e à dinâmica propriamente dita da execução dos programas.

### C. Extensibilidade

*Extensibilidade* é um importante conceito que se refere à capacidade de um sistema incorporar incrementalmente novas funcionalidades, ou novos recursos, por solicitação do usuário, mediante o fornecimento de definições, para as extensões desejadas, indicando-se para tanto a forma como tais extensões operam, em função da combinação dos recursos já disponíveis.

Muito popular entre 1960 e 1975, o conceito de extensibilidade para linguagens de programação prenunciou aquilo que mais tarde viria a se materializar na forma de adaptatividade.

Com a extensibilidade, o programador pode modificar, enquanto codifica, as capacidades da sua linguagem de programação, adequando-a a seus propósitos particulares, tanto em matéria de aspecto externo quanto de funcionalidade.

Não se trata propriamente de uma manifestação de adaptatividade, visto que esta é um fenômeno que ocorre estritamente em tempo de execução, enquanto a extensibilidade é tratada tipicamente na época da compilação.

Essa tendência sofreu um impacto negativo com o advento das novas idéias acerca da maneira de programar, resultantes das pesquisas iniciadas nos anos 1970, tendo como consequência uma abrupta redução dos esforços da comunidade nas pesquisas na área, podendo-se dizer que, em 1975, Standish [1] fez uma documentação histórica dos trabalhos mais importantes previamente publicados sobre o assunto.

Em 1970, em Harvard, Wegbreit publica um relatório técnico sobre linguagens de programação extensíveis [54]. Outros trabalhos significativos nessa área foram o de Hanford e Jones, sobre *sintaxe dinâmica* [66], e o de Cabasino, sobre *linguagens evolutivas e parsers dinâmicos* [65].

Mais recentemente, ocorreu o aparecimento do conceito de *programação extensível*, no qual a idéia da extensibilidade é retomada, porém, agora com algumas regras mais restritivas para sua utilização, que exigem, entre outros, a existência não apenas de uma sintaxe flexível, mas também que todos os ambientes de compilação, de desenvolvimento, de depuração e de execução também respeitem essa exigência. Diversas publicações surgiram nesse sentido. Entre outras, podem-se citar duas, apenas a título de ilustração dessas tendências: em 2001, Bacharach e Playford propõem um *extensor sintático* para a linguagem Java, de certa maneira resgatando valores das linguagens extensíveis [69], e em 2002, Carmi publica um artigo sobre a biblioteca Adapsper [48], projetada para operar como

ambiente básico para o desenvolvimento de *analísadores sintáticos adaptativos* LALR(1), que permitam o tratamento de linguagens extensíveis, ao contrário do que ocorre com os geradores automáticos usuais de analisadores sintáticos.

### D. Adaptatividade

Pode-se considerar que a extensibilidade de linguagens de programação seja uma das mais importantes manifestações na pesquisa da área, sendo tais linguagens anteriores ao aparecimento de trabalhos explicitamente relacionados com o fenômeno da adaptatividade.

O alcance da extensibilidade se limita, entretanto, às componentes da linguagem de programação que podem ser tratadas em tempo de compilação, e corresponde, portanto, às alterações que o usuário pode efetuar sobre a linguagem, adequando-a aos seus particulares propósitos.

Assim, cada usuário pode construir para si uma versão personalizada da linguagem, com a qual possa exprimir a lógica de seus programas da forma mais confortável possível, e numa notação que seja aderente às suas aplicações.

Isso não cobre, no entanto, as necessidades daqueles usuários cujos programas necessitem efetuar alterações dinâmicas, durante a execução, na lógica de seus procedimentos.

Para isso, tais programas necessitariam incorporar recursos de *adaptatividade* e, portanto, dispor de meios para, sem intervenção externa, determinar e realizar alguma automodificação estrutural necessária.

Isso obviamente exige, da parte do ambiente em que a linguagem de programação é processada, a disponibilidade de recursos que permitam a tal *programa adaptativo*, sempre que necessário, automodificar-se de forma consistente.

Esse conceito de adaptatividade, ao qual todo o presente artigo se refere, se estende a qualquer dispositivo cuja operação seja guiada por um conjunto de regras: se tal conjunto for invariável, o dispositivo em questão não será adaptativo, mas se, por seus próprios meios, for ele capaz de modificar o conjunto de regras em questão, o dispositivo será adaptativo.

Em Neto [2], encontra-se uma formalização geral dos *dispositivos adaptativos dirigidos por regras*. Encontram-se extensas informações acerca de trabalhos publicados envolvendo a adaptatividade no sentido acima apresentado (Iwai [32], Pistori [33], Pariente [38], Shutt [45] e Jackson [49]).

### E. Autômatos de topologia variável

Como foi mencionado, e como deixa clara a formulação geral apresentada por Neto em [2], os mais diversos tipos de dispositivos abstratos dirigidos por regras podem ser transformados em dispositivos adaptativos mediante a incorporação de uma *camada adaptativa*, composta dos elementos necessários para que possa ser associado, a cada regra, um conjunto de ações responsáveis pela modificação do próprio conjunto de regras, a ser executada em todas as ocasiões em que a referida regra for acionada, durante a operação do dispositivo.

Um dos dispositivos abstratos que incorporam de forma mais natural esse recurso é o autômato, entendido como qualquer dispositivo com estados cuja operação seja definida por

um conjunto de transições entre esses estados. Cada transição é considerada uma regra, e é possível a cada uma delas associar-se à execução de um conjunto de ações de alteração do conjunto de transições do autômato. O dispositivo adaptativo resultante recebeu vários nomes na literatura e se manifestou de diferentes maneiras no transcorrer da história.

Assim, em 1967, Agasandjan publica, na antiga União Soviética, uma nota de duas páginas, que apresenta a idéia de um tipo de *autômatos com estrutura variável*, que se poderia comparar, em funcionamento e estrutura, a uma forma embrionária de dispositivo adaptativo da classe dos autômatos [68].

Na mesma linha, Salomaa apresenta, em 1968, em trabalho teórico extenso, uma classe de *autômatos finitos com estrutura variável no tempo* [57].

Em 1981, no Brasil, Neto e Magalhães publicavam o primeiro artigo sobre o formalismo dos *autômatos de pilha estruturados* [27], os quais foram usados mais tarde como base para os autômatos adaptativos. Esse trabalho constituiu o alicerce de uma frutuosa prática de ensino da construção de compiladores dirigidos por sintaxe, cujos reconhecedores sintáticos subjacentes são os autômatos de pilha estruturados.

Em 1986, Krithvasan publica seu primeiro trabalho formalizando os *autômatos finitos variantes no tempo* [55], em continuidade às idéias precursoras publicadas por Agasandjan [68] e por Salomaa [57]. Em 1988, Krithvasan estende esses estudos, formalizando de maneira similar os *autômatos de pilha variantes no tempo* [56].

Em 1988, Neto apresentou uma publicação [43] das primeiras idéias do princípio de funcionamento do formalismo reconhecedor, que viria a ser chamado mais tarde de *autômato adaptativo*.

Motivado por problemas de reconhecimento sintático de linguagens de programação, esse artigo apresenta as primeiras idéias do uso de automodificação do autômato como forma de trabalhar sintaticamente elementos de dependências de contexto de uma linguagem de programação, tais como a verificação de tipos e a garantia da utilização exclusiva de variáveis que tenham sido previamente declaradas.

Posteriormente, em 1993, tal idéia viria a permear todo o conteúdo da tese de Neto [3], na qual, em extensivo estudo, consolida a formalização de um dispositivo com conjunto dinâmico de regras e que opera como os autômatos de pilha estruturados, porém, com conjunto de transições que se modifica durante a operação do dispositivo. A esse formalismo deu-se o nome simplificado de *autômato adaptativo*.

Nessa tese, é dada uma especial ênfase às aplicações dos autômatos de pilha estruturados adaptativos, os quais são adaptativos à representação de dependências de contexto, bem como à construção de reconhecedores sintáticos completos para linguagens de programação e de seus compiladores.

Nessa publicação, além da apresentação do formalismo adaptativo, apresenta-se também uma técnica prática e eficiente para a implementação automática de bons reconhecedores sintáticos livres de contexto, usando-se autômatos de pilha estruturados.

Mostra-se ainda como construir autômatos adaptativos que

unificam o tratamento dos aspectos léxicos, sintáticos livres de contexto e sintáticos dependentes de contexto.

Para isso, diversos aspectos da compilação de linguagens de programação imperativas são extensivamente tratados com o emprego do formalismo proposto: análise léxica, declarações e utilização de nomes, escopos aninhados, declaração e verificação de tipos, definição e uso de macros paramétricas etc. Tabelas de símbolos e tabelas de palavras reservadas são eliminadas e substituídas por mecanismos adaptativos equivalentes.

Esse trabalho constitui uma referência bastante completa acerca do formalismo dos autômatos adaptativos e da sua aplicação à implementação de linguagens de programação.

Em 1994, Neto publica na ACM SIGPLAN Notices uma pequena síntese do formalismo dos autômatos adaptativos apresentado em sua tese, síntese essa que deu a conhecer à comunidade internacional o autômato adaptativo [2].

Ainda em 1994, no Worcester Polytechnic Institute, Shutt complementa o trabalho iniciado para gramáticas em sua tese [45], passando então a abranger formalismos de autômatos. Rubinstein e Shutt publicam então dois artigos, nos quais apresentam os seus *autômatos finitos automodificáveis* [46]. Logo em seguida, em 1995, publicam outro artigo, introdutório, sobre os mesmos autômatos finitos automodificáveis [47].

Em continuidade à tese de Neto, sobre métodos de construção de compiladores para linguagens de programação, Pereira apresenta em 1999, em sua dissertação de mestrado [34], uma ferramenta que oferece a seus usuários recursos para a criação e o ensaio de autômatos finitos, de pilha estruturados e adaptativos, o qual se mostrou útil para a implementação de linguagens de programação.

Essa ferramenta proporcionou recursos para a definição formal da sintaxe de linguagens livres de contexto e a produção automática de autômatos de pilha estruturados a partir dessa formalização da linguagem, ou seja, com ela tem-se um gerador automático de reconhecedores sintáticos para linguagens livres de contexto.

Outra face da ferramenta é seu aspecto adaptativo, omitido no tratamento de linguagens regulares e livres de contexto, mas presente na ferramenta, de forma que, havendo necessidade de recursos para o tratamento de dependências de contexto, toda a potência da ferramenta e do formalismo dos autômatos adaptativos pode ser posta em ação.

Em 2001, Neto publicou um texto que teve um papel muito significativo para a área, já que apresenta uma forma generalizada para o fenômeno da adaptatividade, através de uma proposta de formulação geral para os dispositivos formais adaptativos [25].

Segundo essa proposta, todos os dispositivos adaptativos cuja operação possa ser considerada uma alteração dinâmica do conjunto de regras que a definem podem ser expressos de maneira uniforme usando-se o formalismo dos *dispositivos adaptativos dirigidos por regras*, aí apresentados.

Essa arquitetura geral consiste em acrescentar, a um dispositivo subjacente não-adaptativo de qualquer natureza uma camada adaptativa, responsável pela parte dinâmica do con-

junto de regras que definem o dispositivo.

Em 2001, Rocha e Neto apresentaram um estudo conceitual dos autômatos adaptativos, explorando sua aplicabilidade teórica [21]. Em particular, esse artigo contém uma demonstração da abrangência dos autômatos adaptativos, provando sua equivalência com as máquinas de Turing.

Em 1996, Saitou e Jakiela escrevem um artigo em que estudam os *autômatos automontáveis unidimensionais* [70], dispositivos que também exibem adaptatividade e que são criados dinamicamente através da montagem incremental do autômato desejado a partir de um conjunto finito de partes acopláveis disponíveis.

Mais tarde, em 2002, Klein e Kutrib publicaram outro artigo [60], em que apresentam os dispositivos a que denominaram *autômatos finitos automontáveis*.

Em 2003, Pistori, em sua tese de doutorado, faz uma pesquisa bastante cuidadosa dos progressos até então atingidos na área, além de dar diversas outras contribuições técnicas ao campo da adaptatividade e suas aplicações.

Como algumas das contribuições apresentadas em seu trabalho [33], o autor define formalmente o conceito de autômato finito adaptativo, propõe diversas simplificações na notação dos autômatos adaptativos e elimina algumas redundâncias de propostas anteriores.

Em 2003, Pistori e Neto publicam uma documentação sobre a ferramenta AdapTools [35], desenvolvida como parte da tese de Pistori [33], e que tem sido usada como uma das principais referências para a utilização da ferramenta.

Em 2006, Keung e Tyagi publicam um artigo [58] contendo a descrição do conceito de *reconfigurabilidade do espaço de estados* e usam esse conceito em uma proposta de implementação física eficiente dos *autômatos finitos automodificáveis*, de Rubinstein e Shutt [46].

#### F. Gramáticas dependentes de contexto

Em 1965, Van Wijngaarden dava início à sua importante contribuição apresentando pela primeira vez as chamadas *gramáticas de dois níveis*, ou *gramáticas W*, nome dado em sua homenagem, com a publicação do seu artigo sobre o *projeto ortogonal de linguagens de programação* [52].

O autor propõe, nessa publicação, um novo tipo de gramática, através da qual podem ser aplicadas algumas diretrizes inovadoras para o projeto e descrição formal de linguagens de programação, centradas no conceito da ortogonalidade dos componentes das linguagens de programação.

A ortogonalidade foi acompanhada de outra proposta, relativa à maneira de descrever gramaticalmente a linguagem em questão: em lugar das clássicas gramáticas que definem as linguagens através de conjuntos fixos de regras, usa-se uma metagramática, que permite gerar novas regras “sob medida” para cada situação, as quais vão sendo incorporadas ao conjunto de regras já existentes, completando-as de acordo com a necessidade de cada texto em análise.

Mais tarde, tais *gramáticas de dois níveis* seriam utilizadas para a especificação formal da importante linguagem de programação Algol 68, consagrada entre os mais significativos

marcos da evolução das linguagens de programação [53].

A contribuição de Van Wijngaarden foi importantíssima como pioneira na representação estritamente sintática de fenômenos lingüísticos dependentes de contexto, através de métodos gramaticais generativos.

Alguns anos após o trabalho de Wegbreit sobre linguagens de programação extensíveis [54], foi publicado em 1974 o trabalho de Hanford e Jones sobre *sintaxe dinâmica* [66], formalizado como uma aplicação das idéias referentes à adaptatividade ao contexto da definição sintática de linguagens de programação com características extensíveis, reforçando assim a tendência ao aparecimento de uma forma gramatical diferente daquela representada pelas gramáticas de dois níveis, para a formalização de dependências de contexto em linguagens de programação.

Pouco depois, em 1976, foi publicado o relatório revisado da linguagem Algol 68 [53], no qual os autores exploram ao limite os recursos oferecidos pelas gramáticas W, de dois níveis, e aplicam ao projeto dessa linguagem o inovador e importante conceito de ortogonalidade, ambos previamente apresentados pelo autor em [52].

Em 1990, diversos artigos foram publicados em torno da adaptatividade em âmbito gramatical.

Em abril desse ano, Meek publicou seu trabalho [72] centrado na representação das dependências de contexto encontradas nas linguagens de programação. A terminologia empregada dá o nome de *semântica estática* aos fenômenos sintáticos representados pelas *dependências de contexto*. Tal impropriedade foi em novembro apontada por Christiansen [44].

No artigo de maio de 1990, Burshteyn discute o conceito e a prática de alterar a gramática da linguagem de programação durante a análise dos programas nela codificados [64].

Em um segundo trabalho, de novembro do mesmo ano, Burshteyn detém-se em considerações sobre a geração de linguagens formais por *gramáticas modificáveis* e no reconhecimento sintático de programas denotados nas linguagens por elas formalizadas [51].

Esses dois trabalhos muito influenciaram os desenvolvimentos realizados na época, propiciando a compreensão da utilidade de *formalismos automodificáveis* em aplicações voltadas à especificação das dependências de contexto encontradas nas linguagens de programação.

Ainda em 1990, Christiansen publicou um excelente trabalho [44], motivado pelos artigos de Burshteyn [51] e Meek [72], publicados anteriormente no mesmo ano. Já na introdução de seu artigo, comenta a impropriedade terminológica do trabalho de Meek e propõe o uso das técnicas de Burshteyn como forma prática para representar e implementar linguagens contendo dependências de contexto. Em adição, coleta diversas outras publicações então conhecidas sobre *gramáticas adaptáveis* e tece comentários sobre as mesmas.

Em 1992, Cabasino, Paolucci e Todesco publicam um artigo sobre *gramáticas evolutivas e analisadores sintáticos dinâmicos*, contribuindo mais uma vez para a evolução da idéia da adaptatividade em formalismos gramaticais e tecendo considerações acerca da sua realização prática.

Em 1993, Shutt publica sua dissertação de mestrado, sobre *gramáticas adaptáveis recursivas* [45], outra importante contribuição para esses formalismos gramaticais dependentes de contexto, trabalho muito influenciado pelos conceitos de automodificação, contidos nos formalismos similares publicados até então.

No ano de 1994, Boullier publica mais um trabalho sobre *formalismos gramaticais dinâmicos* [50], com os quais o autor propõe uma forma de efetuar não apenas a descrição sintática de uma linguagem de programação, como também a verificação de tipos e outros aspectos das dependências de contexto usualmente encontradas, chegando a ilustrar o uso de sua proposta para a resolução de aspectos bastante complexos, tais como os polimorfismos, tipos derivados, *overloading*, e outras formas não-triviais de dependências de contexto, freqüentes nas linguagens modernas de programação.

Em 2000, Iwai publica sua tese [32], na área dos formalismos gramaticais adaptativos, até então inexplorados no Brasil.

Em sua contribuição, propõe a formalização de linguagens dependentes de contexto através das *gramáticas adaptativas*, que utilizam como base gramáticas livres de contexto, às quais se adicionam, como forma principal de incremento do poder de expressão, mecanismos adaptativos de alteração do seu próprio conjunto de produções.

Adicionalmente, para simplificar o tratamento das dependências de contexto, as gramáticas adaptativas também lançam mão de *símbolos de contexto*, que podem ser associados dinamicamente ao texto reconhecido, em função da identificação de situações particulares de dependência de contexto na sintaxe das cadeias analisadas.

Aycock publica, em 2003, seu artigo de apoio ao uso de um formalismo que denominou *gramáticas dinâmicas generativas*, e que propõe como modelo de computação [62].

Com essa proposta, o autor unifica diversas áreas diferentes de aplicação por meio do emprego desse modelo computacional comum, de caráter gramatical.

Em 2004, Pariente publica sua tese sobre as gramáticas adaptativas com verificação de aparência [38]. Nesse trabalho, é feita uma compilação extensiva de diversos tipos de gramáticas com conjuntos variáveis de regras e são estabelecidos entre elas diversos vínculos e comparações.

Em 2006, Jackson publica seu livro sobre adaptatividade e dependências de contexto em análise sintática [49], que inclui informações sobre uma ferramenta de auxílio ao uso das gramáticas adaptativas denominadas gramáticas- $\xi$  para a representação de linguagens com dependências de contexto.

Trata-se provavelmente do único livro publicado que aborda de forma mais extensiva e abrangente esse assunto da apresentação das dependências de contexto e seu tratamento computacional através de técnicas adaptativas.

### G. Outros formalismos adaptativos

Logo após a publicação da tese em que foram apresentados os autômatos adaptativos [3], iniciaram-se diversas pesquisas em torno do tema da adaptatividade, nas quais se procurou transpor as idéias já utilizadas com autômatos para outros

tipos de dispositivos formais.

Como primeiro resultado completo nesse sentido, Almeida publicou, em 1995, sua tese sobre *statecharts adaptativos* [31], incorporando adaptatividade ao formalismo clássico dos statecharts, de Harel [76]. Apresentou ainda, como complemento, uma ferramenta visual para a criação e simulação de statecharts adaptativos.

Em 1997, Santos acrescentou extensões, dotando com recursos mais expressivos a teoria e a ferramenta apresentadas por Almeida, para permitir que fossem ensaiados aspectos de sincronização explícita entre *statecharts*. Isso se fez através da adição de redes de Petri aos *statecharts*, para a explicitação do sincronismo entre eventos.

Do mesmo modo que em Almeida [31], da pesquisa de Santos [77] também resultou uma ferramenta visual para a definição e ensaio dos seus *statecharts adaptativos sincronizados*.

Outra publicação, que teve um papel muito importante como forma de demonstração prática da aplicabilidade da adaptatividade, foi materializada em 1999, no artigo de Neto e Basseto sobre composição musical automática, baseada em *algoritmos adaptativos* [11].

O formalismo matemático em que tais algoritmos se baseiam é a *rede de Markov adaptativa*, uma espécie de autômato finito estocástico de topologia completa, a cujas transições são associadas probabilidades, sendo estas modificadas dinamicamente de acordo com as transições realizadas. Famílias de tais dispositivos podem ser usadas, permitindo-se que cada um deles possa modificar, segundo regras bem estabelecidas, as probabilidades associadas às próprias transições e também às dos outros dispositivos com que operam.

Um programa muito simples, denominado Lassus [71], elaborado com base nesse princípio, ilustra a eficácia dos dispositivos adaptativos como mecanismos inteligentes a serem usados na elaboração computacional de peças musicais, segundo regras determinadas.

Em 2000, Basseto publica sua dissertação de mestrado [29], na qual são formalizadas as *redes de Markov adaptativas*. Nesse trabalho, o autor explora diversos aspectos da automatização da composição musical, utilizando como linguagem de especificação uma gramática sensível ao contexto, e implementando-a com redes de Markov adaptativas, abrindo caminho para muitos desdobramentos na área da utilização do computador como coadjuvante nesse tipo de processo da criação artística.

Em 2001, ao lado da apresentação da teoria geral dos *dispositivos adaptativos dirigidos por regras* [25], Neto ilustra os conceitos aí definidos com a apresentação das *tabelas de decisão adaptativas* como dispositivo adaptativo obtido pela incorporação de mecanismos adaptativos a um formalismo subjacente representado pelas tabelas de decisão tradicionais, o que viria mais tarde a mostrar-se útil na formulação de sistemas adaptativos de tomada de decisões [9], [33].

Em 2006, Pistori, Neto e Pereira publicam um artigo sobre árvores de decisão adaptativas não-determinísticas [20], muito adequadas para a formulação de tomadas de decisão do tipo diagnóstico inteligente, inferido a partir de sintomas.

#### IV. APLICAÇÕES DA TECNOLOGIA ADAPTATIVA

Inúmeras aplicações potenciais existem para a tecnologia derivada da adaptatividade. Enumeram-se abaixo algumas, que se podem considerar significativas pelo impacto que causaram desde a época em que foram desenvolvidas.

##### A. Inferência

Em 1988, Neto e Iwai publicaram um primeiro artigo sobre inferência [23], no qual é mostrado um método, empregando autômatos adaptativos, que, partindo de amostras positivas e negativas de uma linguagem regular, é capaz de inferir um autômato finito que representa uma boa aproximação do autômato que aceita a linguagem em questão.

Inspirada nesse trabalho, e dando continuidade ao mesmo, em 2006, Matsuno publica, em sua dissertação de mestrado [8], os resultados obtidos aplicando a tecnologia adaptativa à inferência de linguagens regulares e livres de contexto.

##### B. Arte usando computador

Nos anos de 1999 e 2000, Basseto cria, usando tecnologia adaptativa baseada em redes de Markov, uma aplicação muito interessante, na área da geração automática de música por computador, criada com base em regras bem estabelecidas [11], [29], [71].

##### C. Processamento de textos em linguagem natural

Em 2000, a área do processamento de linguagens naturais ganha as contribuições de Menezes [30], com a criação de um método inteligente de etiquetagem automática de textos em língua natural.

Essa pesquisa apresentou, como um de seus produtos, um método, baseado em autômatos adaptativos, de análise morfológica para a língua portuguesa.

Em um protótipo, o autor desenvolveu um coletor de palavras usando um autômato adaptativo enriquecido com mecanismos de avaliação da frequência de utilização dos caminhos, de forma que dinamicamente o dispositivo efetua suas buscas percorrendo o caminho até então mais utilizado, ou seja, o de maior frequência constatada.

O dispositivo explora a idéia de implementar o classificador para operar em duas etapas: a primeira, de aprendizagem, feita sobre um corpus etiquetado, no qual as palavras e as respectivas etiquetas são coletadas e aprendidas; e a segunda, de utilização dos fatos coletados, na qual um texto não etiquetado é analisado e suas palavras, classificadas de acordo com o que tenha sido aprendido na fase anterior.

Em 2001, Taniwaki apresenta sua dissertação [39], na qual mostra a viabilidade da representação de fenômenos lingüísticos complexos da linguagem natural, usando para isso formalismos adaptativos. Mostra ainda a equivalência entre tais formalismos e outros tradicionalmente empregados na análise e processamento de linguagem natural, tais como ATN e DCG.

Após a experiência de Menezes [30], optou-se por investir na pesquisa da parte sintática das linguagens naturais, em particular, nos seus aspectos dependentes de contexto, e como

primeiro resultado nessa direção, Neto e Moraes publicaram, em 2003, um material [5] apresentando uma técnica de descrição de dependências de contexto usando como base o mecanismo formal oferecido pelas gramáticas adaptativas, propostas por Iwai em [32].

##### D. Síntese de voz

Em 2004, é publicado também um artigo sobre a aplicação da tecnologia adaptativa à síntese de voz [16]. Esse foi um dos primeiros desdobramentos de um dos experimentos conduzidos na tese de Pistori [33], e utiliza autômatos adaptativos para escolher a composição apropriada de fonemas pré-gravados, com base em regras de articulação vocal aplicadas a textos escritos em português.

##### E. Reconhecimento de padrões

Em 2002, Costa, Hirakawa e Neto publicam um artigo sobre o uso de mecanismos baseados em autômatos adaptativos no reconhecimento automático de padrões geométricos, usando pela primeira vez essa tecnologia em robótica [24].

Segue-se a publicação [33] de Pistori, na mesma direção, em 2003, explorando em profundidade essa aplicação, orientando-a para padrões mais complexos e variáveis e apresentando diversas aplicações da tecnologia adaptativa: reconhecimento de formas, síntese de voz e elaboração de interfaces humano-computador.

Diversas das propostas apresentadas nessa tese dariam mais tarde origem a três projetos de aplicação da tecnologia adaptativa, em desenvolvimento na UCDB – Universidade Católica Dom Bosco, em Campo Grande – MS.

O projeto SIGUS [78], ora em curso, originou-se do tema do artigo publicado em 2004 por Pistori e Neto [7] sobre os primeiros resultados do uso de tecnologia adaptativa para o reconhecimento de padrão, em particular, correspondente à identificação dos símbolos da linguagem de sinais. Outros dois projetos, atualmente também em andamento, se orientaram, respectivamente, à detecção automática [10] de defeitos em peles e couros bovinos – DTCOURO [80] – e a um sistema de apoio à análise automática de comportamento animal em laboratório – TOPOLINO [79].

##### F. Tomada de decisão

Após a publicação, em 2001, do artigo de Neto, apresentando as tabelas de decisão adaptativas, outros trabalhos surgiram explorando a adaptatividade na tomada de decisões.

Assim, em 2003, Pistori desenvolveria, a partir das tabelas de decisão adaptativas, os conceitos de árvores de decisão adaptativas, as quais, pela aderência conceitual a certos tipos de tomadas de decisão, tornam-se um formalismo muito interessante para uso nessas aplicações [33].

Em 2005, Pedrazzi, Tchemra e Rocha contribuíram com um artigo sobre o uso de *tabelas de decisão adaptativas* em aplicações à resolução de problemas de tomada automática de decisão [9].

No artigo de 2006 de Pistori, Neto e Pereira [20], são conceituadas as *árvores de decisão adaptativas* e ilustra-se sua aplicação à tomada de decisão em diagnósticos automáticos

com base em sintomas.

### G. Linguagens de programação adaptativas

Em 2003, Rocha e Neto publicam um primeiro trabalho sobre a inclusão de características de adaptatividade em linguagens de programação [6], em particular, linguagens funcionais, mostrando a viabilidade de tal empreendimento através da apresentação de uma possível arquitetura.

Em 2006, Freitas e Neto contribuem com um artigo sobre *linguagens de programação adaptativas* [15] e nesse trabalho discutem o novo estilo de programação daí derivado.

### H. Otimização de código

Em 2004, Luz publica em sua dissertação de mestrado [22] uma aplicação da tecnologia adaptativa à otimização de código em compiladores, empregando uma forma de otimização do tipo *peephole*, aperfeiçoada pela inclusão da adaptatividade nos algoritmos de otimização utilizados.

### I. Meta-modelagem

Em 2004, Camolesi e Neto publicam as primeiras idéias da modelagem adaptativa [18] e, com base nelas, em 2007, Camolesi apresenta sua tese de doutorado [36], na qual a tônica é a proposta de um metambiente, materializada por meio de uma ferramenta com interface visual, destinada a oferecer ao usuário meios para a construção automática de ambientes completos para a definição, formalização, simulação e ensaios de dispositivos formais, adaptativos ou não, especificados pelo próprio usuário.

### J. Computação evolutiva

Em 2005, Pistori, Martins e Castro publicaram um artigo confrontando algoritmos genéticos com autômatos finitos adaptativos, em uma aplicação na qual foram comparadas e correlacionadas a adaptação dos indivíduos e a evolução da população [19].

Em 2007, Bravo et al. publicam um artigo em que se aplica a adaptatividade para simular algoritmos genéticos, como parte de um projeto envolvendo a aplicação da tecnologia adaptativa ao estudo da biodiversidade [4].

### K. Engenharia de software

Também em 2005, Silva e Neto apresentaram um artigo [13] propondo um esquema para o projeto de linguagens para a especificação de software, que corresponde a uma primeira aproximação entre a tecnologia adaptativa e a engenharia de software.

### L. Robótica

Em continuidade ao trabalho de Costa, Hirakawa e Neto em 2002, sobre reconhecimento de padrões geométricos [24], no mesmo evento de 2005, Souza e Hirakawa publicam [14] outra contribuição da adaptatividade à robótica, envolvendo o mapeamento automático e a navegação em ambientes desconhecidos usando como formalismo de representação os autômatos adaptativos.

## V. PROJETOS, EVENTOS E OUTROS RECURSOS

Em adição às publicações dos diversos autores, na forma de textos técnicos, outros esforços têm sido realizados, na forma de projetos técnicos, eventos, grupos de trabalho e páginas web, entre outros.

Assim, podem ser mencionados os projetos citados anteriormente, que aplicam a tecnologia adaptativa no reconhecimento de padrões [78], [79], [80], bem como outro projeto, relacionado com o estudo de biodiversidade, no qual a tecnologia adaptativa está sendo ensaiada como alternativa para algoritmos genéticos [4] e empregada em tomadas de decisão.

Merecem menção também alguns eventos dedicados à adaptatividade nos quais alguns dos artigos antes referenciados foram publicados: ICANNGA 2005 [12] e I WTA [67].

Alguns recursos adicionais sobre adaptatividade podem ser encontrados na Internet, nas páginas Web de grupos de pesquisa e em materiais didáticos sobre adaptatividade, por eles disponibilizados [17], [59], [61], [73].

## VI. CONCLUSÃO

O presente artigo faz um levantamento, relativamente abrangente, das principais contribuições encontradas na literatura nacional e internacional ao desenvolvimento dos conceitos e das aplicações ligadas à adaptatividade.

Procurou-se rastrear os caminhos percorridos nesse sentido, partindo das manifestações mais antigas do conceito e registrando a forma como os diversos assuntos a ele relacionados foram sendo tratados e desenvolvidos ao longo dos anos.

Todas essas contribuições acabaram criando para a adaptatividade um conjunto de poderosos recursos, de natureza não apenas conceitual e teórica, mas também de caráter aplicativo, na forma de métodos, técnicas e ferramentas, que foram sendo capitalizados ao longo dessa história.

Isso torna notório o poder da adaptatividade e o imenso potencial apresentado pelos dispositivos adaptativos como ferramenta para a resolução de problemas extremamente complexos e variados.

Como ferramenta de apoio ao desenvolvimento de aplicações oriundas de uma grande diversidade de áreas de interesse, a tecnologia adaptativa se tem demonstrado, como pode ser observado na literatura apresentada, uma poderosa colaboradora das técnicas tradicionais empregadas pela Engenharia de Software e pela Inteligência Artificial.

Espera-se que a animadora visão panorâmica aqui apresentada atraia, cada vez mais, pesquisadores de variadas origens a explorarem, em suas pesquisas e aplicações, o poderoso instrumental representado pela adaptatividade e pelos variados dispositivos adaptativos já estudados, além de encorajá-los a desenvolver também seus próprios dispositivos adaptativos, conforme seus interesses e necessidades particulares.

Espera-se, enfim, que a bibliografia aqui referenciada, longe de esgotar a literatura publicada sobre o assunto, venha a auxiliar os futuros interessados a estudarem e conhecerem mais profundamente essa área, e a enriquecerem esse fascinante domínio do conhecimento com suas próprias contribui-

ções.

#### AGRADECIMENTO

Ao Dr. César Alberto Bravo Pariente, pelo valioso auxílio prestado na coleta bibliográfica. À Escola Politécnica da USP e ao PCS, por manterem o LTA e pelo seu apoio à realização do I WTA – Primeiro Workshop de Tecnologia Adaptativa.

#### REFERÊNCIAS

- [1] STANDISH T.A. *Extensibility in Programming Language Design* ACM SIGPLAN Notices, p. 18-21, vol 10 n. 7, July, 1975.
- [2] NETO, J.J. *Adaptive Automata for Context-Sensitive Languages*. SIGPLAN NOTICES, Vol. 29, n. 9, September, 1994, pp. 115-124.
- [3] NETO, J.J. *Contribuições à Metodologia de Construção de Compiladores*. Tese de Livre Docência, Escola Politécnica da USP, São Paulo, 1993.
- [4] BRAVO, C. et al. *Towards an Adaptive Implementation of Genetic Algorithms* INBI 2007, XXXIII CLEI – Conferencia Latinoamericana de Informática, San José, Costa Rica, 2007.
- [5] NETO, J.J.; MORAES, M. *Using Adaptive Formalisms to Describe Context-Dependencies in Natural Language*. Lecture Notes in Artificial Intelligence. Mamede, N.J., Baptista, J., Trancoso, I., Nunes, M.G.V. (Eds.): Computational Processing of the Portuguese Language 6th International Workshop, PROPOR 2003, Volume 2 721, pp 94-97, Faro, Portugal, Springer-Verlag, 2003.
- [6] ROCHA, R.L.A.; NETO, J.J. *Uma proposta de linguagem de programação funcional com características adaptativas*. IX Congreso Argentino de Ciencias de la Computación. La Plata, Argentina, Octubre de 2003.
- [7] PISTORI, H.; NETO, J.J. *An Experiment on Handshape Sign Recognition using Adaptive Technology: Preliminary Results*. XVII Brazilian Symposium on Artificial Intelligence – SBIA'04. São Luis, September 29 – October 1, 2004.
- [8] MATSUNO, I.P. *Um Estudo do Processo de Inferência de Gramáticas Regulares e Livres de Contexto Baseados em Modelos Adaptativos*. Dissertação de Mestrado, USP, São Paulo, 2006.
- [9] PEDRAZZI, T., TCHEMRA, A. H., ROCHA, R. L. A. *Adaptive Decision Tables – a Case Study of their Application to Decision-Taking Problems*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANNGA 2005, Coimbra, March 21-23, 2005.
- [10] PISTORI, H. et al. *Defect Detection in Raw Hide and Wet Blue Leather* CompIMAGE – Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications, Coimbra, 2006.
- [11] NETO, J.J. e BASSETO, B. A. *A Stochastic Musical Composer Based on Adaptive Algorithms*. Proceedings of the 6th Brazilian Symposium on Computer Music - SBC&M99, Rio de Janeiro, 1999.
- [12] RIBEIRO, B. et al (ed.) *Adaptive and Natural Computing Algorithms – Proceedings of the International Conference in Coimbra, Portugal, 2005*. ISBN 3-211-24934-6 Springer Computer Science series, Springer – Wien New York.
- [13] SILVA, P. S. M., NETO, J. J. *An Adaptive Framework for the Design of Software Specification Languages*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms - ICANNGA 2005, Coimbra, March, 2005.
- [14] SOUSA, M.A.A., HIRAKAWA, A.H. *Robotic Mapping and Navigation in Unknown Environments Using Adaptive Automata*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANNGA 2005, Coimbra, March, 2005.
- [15] FREITAS, A.V.; NETO, J.J. *Adaptive Languages and a New Programming Style* 6th WSEAS International Conference on Applied Computer Science (ACS'06) – Tenerife, Canary Islands, Spain, December, 2006.
- [16] ZUFFO, F., PISTORI, H. *Tecnologia Adaptativa e Síntese de Voz: Primeiros Experimentos* Anais do V Workshop de Software Livre – WSL. Porto Alegre, junho de 2004.
- [17] [http://www.pcs.usp.br/~lta/disciplinas/fun\\_apl\\_tec\\_ada.htm](http://www.pcs.usp.br/~lta/disciplinas/fun_apl_tec_ada.htm) – página da disciplina *Fundamentos e Aplicações da Tecnologia Adaptativa*.
- [18] CAMOLESI, A.R., NETO, J.J. *Modelagem Adaptativa de Aplicações Complexas*. XXX Conferencia Latinoamericana de Informática – CLEI'04. Arequipa - Peru, Setiembre 27 - Octubre 1, 2004.
- [19] PISTORI, H., MARTINS, P.S., CASTRO Jr., A.A. *Adaptive Finite State Automata and Genetic Algorithms – Merging Individual Adaptation and Population Evolution*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANNGA 2005, Coimbra, March, 2005.
- [20] PISTORI, H.; NETO, J.J.; PEREIRA, M.C. *Adaptive Non-Deterministic Decision Trees: General Formulation and Case Study*. INFOCOMP Journal of Computer Science, Lavras, MG, 2006.
- [21] ROCHA, R.L.A.; NETO, J.J. *Autômato Adaptativo, Limites e Complexidade em Comparação com Máquina de Turing*. In: Proceedings of the second Congress of Logic Applied to Technology – LAPTEC'2000. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, p. 33-48, 2001.
- [22] LUZ, J.C. *Tecnologia Adaptativa Aplicada à Otimização de Código em Compiladores*. Dissertação de Mestrado, EPUSP, São Paulo, 2004.
- [23] NETO, J.J.; IWAI, M.K. *Adaptive Automata for Syntax Learning*. CLEI 98 – XXIV Conferencia Latinoamericana de Informática, MEMORIAS. pp. 135-149, Quito, Equador, 1998.
- [24] COSTA, E.R.; HIRAKAWA, A.R.; NETO, J.J. *An Adaptive Alternative for Syntactic Pattern Recognition*. Proceeding of 3rd International Symposium on Robotics and Automation, ISRA 2002, , pp. 409-413. Toluca, Mexico, 2002.
- [25] NETO, J.J. *Adaptive Rule-Driven Devices – General Formulation and Case Study* Lecture Notes in Computer Science, Springer-Verlag, Vol. 2494, pp. 234-250. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Pretoria, South Africa, July 23-25, 2001.
- [26] NETO, J.J. *Introdução à Compilação*. Editora LTC, Rio de Janeiro, 1987.
- [27] NETO, J.J. e MAGALHÃES, M.E.S. *Reconhecedores Sintáticos - Uma Alternativa Didática para Uso em Cursos de Engenharia*. XIV Congresso Nacional de Informática, pp. 171-181, São Paulo, 1981.
- [28] NETO, J.J.; PARIENTE, C.B.; LEONARDI, F. *Compiler Construction – a Pedagogical Approach*. Proceedings of the V International Congress on Informatic Engineering – ICIE 99, Buenos Aires, Argentina, 1999.
- [29] BASSETO, B.A. *Um Sistema de Composição Musical Automatizada, Baseado em Gramáticas Sensíveis ao Contexto, Implementado com Formalismos Adaptativos*. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2000.
- [30] MENEZES, C.E.D. *Um Método para a Construção de Analisadores Morfológicos, Aplicado à Língua Portuguesa, Baseado em Autômatos Adaptativos*. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2000.
- [31] ALMEIDA Jr., J.R. *Stad: uma Ferramenta para Representação e Simulação de Sistemas Através de Statecharts Adaptativos*. Tese de Doutorado, Escola Politécnica da USP, São Paulo, 1995.
- [32] IWAI, M.K. *Um Formalismo Gramatical Adaptativo para Linguagens Dependentes de Contexto*. Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2000.
- [33] PISTORI, H. *Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações*. Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2003.
- [34] PEREIRA, J.C.D. *Ambiente Integrado de Desenvolvimento de Reconhecedores Sintáticos, Baseado em Autômatos Adaptativos*. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 1999.
- [35] PISTORI, H.; NETO, J.J. *AdapTools: Aspectos de Implementação e Utilização* Boletim Técnico PCS, Escola Politécnica, São Paulo, 2003.
- [36] CAMOLESI, A.R. *Proposta de um Gerador de Ambientes para Modelagem de Aplicações Usando Tecnologia Adaptativa* – Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2007.
- [37] PAGAN, F.G. *Formal Specification of Programming Languages: a Panoramic Primer*, Prentice-Hall 1981.
- [38] PARIENTE, C.A.B. *Gramáticas Livres de Contexto Adaptativas com Verificação de Aparência*. Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2004.
- [39] TANIWAKI, C.Y.O. *Formalismos Adaptativos na Análise Sintática de Linguagem Natural* Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2001.

- [40] McGETTRICK, A.D. *The Definition of Programming Languages* Cambridge Computer Science Series Texts – 11, Cambridge University Press, 1980.
- [41] SLONNEGER, K. and KURTZ, B.L. *Formal Syntax and Semantics of Programming Languages – a Laboratory Based Approach* Addison Wesley, 1995.
- [42] NETO, J.J. *Autômatos em Engenharia de Computação – Uma Visão Unificada* Primeira Semana de Ciencia y Tecnologia de la Sociedad Chotana de Ciencias y la Red Mundial de Científicos Peruanos, Junio 22-27, Ciudad de Chota, Perú, 2003.
- [43] NETO, J.J. *Uma Solução Adaptativa para Reconhedores Sintáticos*. Anais EPUSP – Engenharia de Eletricidade – série B, vol. 1, pp. 645-657, São Paulo, 1988.
- [44] CHRISTIANSEN, H. *A survey of adaptable grammars*. SIGPLAN Notices, vol.25, n.11, p.35-44, 1990.
- [45] SHUTT, J.N. *Recursive Adaptable Grammars*, M.S. Thesis, Computer Science Department, Worcester Polytechnic Institute, Worcester Massachusetts, 1993. (Emended 16 December 2003.)
- [46] RUBINSTEIN, R.S.; SHUTT, J.N. *SMFA: Self-Modifying Finite Automata* – IFIP Congress, Vol. 1 pages 493–498, 1994.
- [47] RUBINSTEIN, R.S.; SHUTT, J.N. *Self-Modifying Finite Automata: an Introduction*, Information Processing Letters, v.56 n.4, p.185-190, Nov. 24, 1995.
- [48] CARMÍ, A. *Adapser: An LALR(1) Adaptive Parser*, The Israeli Workshop on Programming Languages & Development Environments, Haifa, Israel, 1 July 2002.
- [49] JACKSON, Q.T., *Adapting to Babel: Adaptivity and Context-Sensitivity in Parsing*, Ibis Publications, Plymouth, Massachusetts, March 2006.
- [50] BOULLIER, P. *Dynamic Grammars and Semantic Analysis*, INRIA Research Report 2322, August 1994.
- [51] BURSHTEYN, B. *Generation and Recognition of Formal Languages by Modifiable Grammars*, ACM SIGPLAN Notices 25 no. 12 pp. 45-53. 1990.
- [52] WIJNGAARDEN, A. *Orthogonal Design and Description of Formal Languages*. Mathematisch Centrum Amsterdam, MR 76, 1965.
- [53] WIJNGAARDEN, A. et al. *Revised Report on the Algorithmic Language Algol 68*, 1973 – Springer-Verlag 1976.
- [54] WEGBREIT, B. *Studies in Extensible Programming Languages* Technical Report ESD-TR-70-297. Harvard University, Cambridge, MA, 1970.
- [55] KRITHVASAN, K. *Time Variant Finite Automata*. International Journal of Computer Mathematics, Vol 19, pp. 103-123. 1986.
- [56] KRITHVASAN, K. *Time Variant Pushdown Automata*. International Journal of Computer Mathematics, Vol 24, pp. 223-236. 1988.
- [57] SALOMAA, A. *On Finite Automata with a Time-Variant structure*. Information and Control, 13, pp 85-98, 1968.
- [58] KEUNG, K.M.; TYAGI, A. *State Space Reconfigurability: An Implementation Architecture for Self Modifying Finite Automata* - Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems 2006, Seoul, Korea October, 2006.
- [59] <http://web.cs.wpi.edu/~jshutt/adapt/adapt.html> – página web sobre Gramáticas Adaptativas.
- [60] KLEIN, A.; KUTRIB, M. *Self-Assembling Finite Automata*. COCOON 2002: 310-319 Proceedings. Lecture Notes in Computer Science 2387 Springer-Verlag, 2002.
- [61] [http://en.wikipedia.org/wiki/Adaptive\\_grammar](http://en.wikipedia.org/wiki/Adaptive_grammar) – página web sobre Gramáticas Adaptativas.
- [62] AYCOCK, J. *A Program Execution Model Based on Generative Dynamic Grammars* – IASTED CST 2003 – Cancun, México 2003.
- [63] DI FIORINO, A.C. *Some Remarks on the Syntax of Symbolic Programming Languages*, Communications of the ACM, 6(8), 1963, 456–460.
- [64] BURSHTEYN, B. *On the Modification of the Formal Grammar at Parse Time* ACM SIGPLAN Notices Vol. 25, No. 5, pp. 117-123, 1990.
- [65] CABASINO, S.; PAOLUCCI, P.S.; TODESCO, G.M. *Dynamic Parsers and Evolving Grammars*, ACM SIGPLAN Notices, 27(11), 39-48, 1992.
- [66] HANFORD, K.V.; JONES, C.B. *Dynamic Syntax: A Concept for the Definition of the Syntax of Programming Languages*, Annual Review in Automatic Programming, volume 7, 115–142. 1974.
- [67] <http://www.pcs.usp.br/~lta/iwta/index.html> *I WTA – Primeiro Workshop de Tecnologia Adaptativa*, Escola Politécnica da USP, São Paulo, janeiro de 2007.
- [68] AGASANDJAN, G.A. *Automata with a Variable Structure*, Dokl. Akad. Nauk SSSR, vol. 174, pp. 529-530, 1967.
- [69] BACHARACH, J.; PLAYFORD, K. *The Java Syntactic Extender*; see <http://www.ai.mit.edu/~jrb/jse/jse.pdf>. 2001.
- [70] SAITOU, K.; JAKIELA, M.J. *On Classes of One-Dimensional Self-Assembling Automata* Complex Systems 10, 391-416. 1996.
- [71] BASSETO, B.A. *Lassus* (Compositor Musical de Corais Tonais a Quatro Vozes). <http://www.pcs.usp.br/~lta/download/lassus.zip>. 1999.
- [72] MEEK, B. *The Static Semantics File*, ACM SIGPLAN Notices Vol. 25, No. 4, pp. 33-42, 1990.
- [73] [www.pcs.usp.br/~lta](http://www.pcs.usp.br/~lta) *LTA – Laboratório de Linguagens e Tecnologia Adaptativa da Escola Politécnica da Universidade de São Paulo*.
- [74] CONWAY, M.E. *Design of a Separable Transition-Diagram Compiler* Communications of the ACM vol 6 n. 7, pp. 396-408. July, 1963.
- [75] LOMET, D.B. *A Formalization of Transition Diagram Systems* Journal of the Association for Computing Machinery, Vol. 20, No. 2, pp. 235-267, April 1973.
- [76] HAREL, D. *Statecharts: A visual formalism for complex systems* Science of Computer Programming Volume 8, Issue 3. Pages: 231 – 274. June 1987.
- [77] SANTOS, J.M.N. *Um Formalismo Adaptativo com Mecanismo de Sincronização para Aplicações Concorrentes*. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 1997.
- [78] <http://www.gpec.ucdb.br/sigus> *SIGUS – Plataforma de Apoio ao Desenvolvimento de Sistemas para Inclusão Digital de Pessoas com Necessidades Especiais* (UCDB – Campo Grande – MS).
- [79] [www.gpec.ucdb.br/topolino](http://www.gpec.ucdb.br/topolino) *TOPOLINO – Sistema de Apoio à Análise Automática de Comportamento Animal em Laboratório* (UCDB – Campo Grande – MS).
- [80] [www.gpec.ucdb.br/dtcouro](http://www.gpec.ucdb.br/dtcouro) *DTCOURO – Detecção Automática de Defeitos em Peles e Couros Bovinos* (UCDB – Campo Grande – MS).
- [81] RICCHETTI, P.M. *Geração de Analisadores Sintáticos Baseados em Transdutores a partir de Especificações Gramaticais* Dissertação de Mestrado. Escola Politécnica da USP. São Paulo, 2005.



**João José Neto** é graduado em Engenharia de Eletricidade (1971), mestre em Engenharia Elétrica (1975), doutor em Engenharia Elétrica (1980) e livre-docente (1993) pela Escola Politécnica da Universidade de São Paulo. Atualmente, é professor associado da Escola Politécnica da Universidade de São Paulo e coordena o LTA – Laboratório de Linguagens e Tecnologia Adaptativa do PCS – Departamento de Engenharia de Computação e Sistemas Digitais da EPUSP. Tem experiência na área de Ciência da Computação, com ênfase nos Fundamentos da Engenharia da Computação, atuando principalmente nos seguintes temas: dispositivos adaptativos, tecnologia adaptativa, autômatos adaptativos, e em suas aplicações à Engenharia de Computação, particularmente em sistemas de tomada de decisão adaptativa, análise e processamento de linguagens naturais, construção de compiladores, robótica, ensino assistido por computador, modelagem de sistemas inteligentes, processos de aprendizagem automática e inferências baseadas em tecnologia adaptativa.