

# Agentes adaptativos reativos: formalização e estudo de caso

R. L. Stange, P. R. M. Cereda e J. José Neto

**Resumo**—Um dos objetivos da Inteligência Artificial é construir agentes. No contexto deste trabalho, um agente caracteriza-se como qualquer dispositivo que recebe estímulos de um ambiente e age sobre este ambiente para realização de uma tarefa. Existem diferentes teorias, arquiteturas e linguagens de agentes, com o objetivo de fazê-los desempenhar melhor suas tarefas. Este artigo apresenta uma formulação para um agente reativo com capacidade de automodificação, chamado de *agente reativo adaptativo*. Para isso, foi aplicado o conceito de adaptatividade ao agente, permitindo que este mude o seu comportamento baseado nas suas percepções. Para avaliar o desempenho do agente adaptativo, foi utilizado um estudo de caso para o problema do mundo de wumpus. Os resultados foram comparados a outros tipos de agentes, sendo o agente adaptativo mais vantajoso em alguns aspectos, tais como simplicidade e eficiência.

**Palavras-chave**:—dispositivos adaptativos, agentes inteligentes, inteligência artificial

## I. INTRODUÇÃO

Agentes e sistemas multiagentes representam uma forma de analisar, projetar e implementar sistemas complexos de software. Pesquisadores e desenvolvedores de diferentes disciplinas têm abordado questões relacionadas a agentes, tais como na Inteligência Artificial [1], Programação Orientada a Objetos [2] e Interface Humano-Computador [3].

Assim como outros conceitos de computação, a definição de agentes sofre por não possuir um consenso universalmente aceito [4], [5]. Este trabalho tem o enfoque na Inteligência Artificial e adota, portanto, a definição e classificação de agentes baseadas em [1]. Russell e Norvig [1] definem um *agente* simplesmente como “algo que age”. De acordo com esta definição, é possível vislumbrar um programa de computador como um agente. Entretanto, existem cenários nos quais é esperado que um agente computacional seja racional e opere de forma autônoma, percebendo seu ambiente e eventualmente sendo suscetível a mudanças. Considerando aspectos de racionalidade, um *agente racional* é aquele que realiza uma tarefa que considera correta de acordo com sua lógica, isto é, este age para alcançar os melhores resultados (ou o melhor resultado esperado) [1]. O estudo de *agentes inteligentes* é um dos temas centrais da Inteligência Artificial e seus conceitos podem ser aplicados desde sistemas relativamente pequenos (por exemplo, filtros de e-mail) até sistemas complexos (por exemplo, controle de tráfego aéreo) [4].

A *adaptatividade* é a característica atribuída ao comportamento automodificável de sistemas computacionais que ocorre

Os autores podem ser contactados através dos seguintes endereços de correio eletrônico: rlstange@usp.br, paulo.cereda@usp.br e jjneto@usp.br.

em decorrência de estímulos de entrada e ao histórico de operação desses sistemas [6]. As pesquisas em adaptatividade investigam soluções para diversos problemas da computação, incluindo tomadas de decisão e aprendizado de máquina [7], [8] e processamento de linguagem natural [9]. O objetivo deste trabalho é aplicar adaptatividade em agentes, de forma que estes possam obter melhores resultados na execução de suas tarefas, sem alterar suas propriedades e características originais.

Este artigo está organizado da seguinte forma: a Seção II introduz o conceito de agentes, enfatizando a forma reativa. Posteriormente, aborda a estrutura de agentes, apresentando a organização elementar de projeto de agentes reativos utilizada neste trabalho. A proposta de um agente reativo com a incorporação da adaptatividade é formulada na Seção III. A Seção IV apresenta experimentos para a avaliação de um agente adaptativo reativo utilizando problema do mundo de wumpus como estudo de caso. As considerações finais sobre os resultados obtidos com a incorporação da adaptatividade em uma agente reativo estão na Seção V.

## II. ESTRUTURA DE AGENTES

Um *agente* é definido como um dispositivo que percebe seu ambiente através de *sensores* e age sobre esse ambiente através de *atuadores* [1], conforme ilustra a Figura 1. A percepção refere-se às entradas perceptivas do agente e uma *sequência de percepções* consiste no histórico de todas as percepções até então. A escolha de uma ação pode depender tão somente da percepção atual ou da sequência de percepções obtida até um dado instante no tempo.

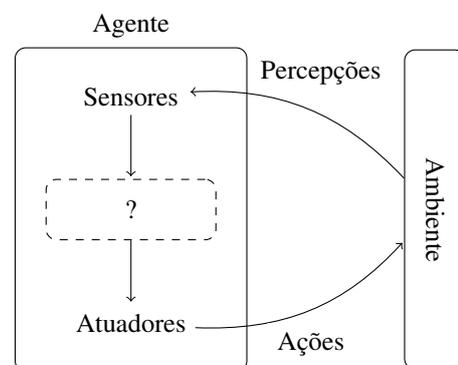


Figura 1. Agentes interagem com ambientes [Adaptada de [1]]

Em termos matemáticos, o comportamento do agente é dado

pela *função do agente*, que realiza o mapeamento de qualquer sequência de percepções em uma ação correspondente. O *programa do agente* é a implementação concreta desta função. O conjunto de sensores e atuadores necessários para que este programa seja executado em um dispositivo computacional é chamado de *arquitetura*. Assim, pode-se afirmar que um agente é composto por um programa e uma arquitetura.

Russell e Norvig [1] apresentam quatro tipos básicos de programas de agentes que possuem princípios subjacentes a quase todos os sistemas inteligentes:

- 1) *Agente reativo simples*: age em função da percepção atual, ignorando o histórico de percepções.
- 2) *Agente reativo baseado em modelo*: conserva um estado interno que depende do histórico de percepções e utiliza este modelo para tomar decisões.
- 3) *Agente baseado em objetivos*: utiliza o conceito da definição de objetivos a serem alcançados, de tal forma que o agente combina o modelo interno, idêntico ao agente baseado em modelos, com informações acerca dos objetivos para escolher as ações que levem a atingi-los.
- 4) *Agente baseado em utilidade*: utiliza um termo chamado *utilidade* para mensurar o quão satisfatória foi a solução encontrada para atingir um objetivo. A função de utilidade é essencialmente a internalização da medida de desempenho do agente.

É importante destacar que os quatro tipos básicos de programas de agentes apresentados compartilham elementos comuns.

#### A. Agentes reativos

A função do *agente reativo* é simplesmente realizar o mapeamento de uma percepção de entrada em uma ação de saída. Em diversas aplicações na área de robótica, a ausência de um estado interno que representa o histórico das percepções pode ser limitante; em outras áreas, como jogos, tal ausência não é impactante e pode tornar o agente mais eficiente [10].

A vantagem dos agentes reativos concentra-se no fato de que as regras do tipo  $\langle \text{condição} \mapsto \text{ação} \rangle$  fornecem uma representação inteligível, modular e eficiente. Porém, não há uma representação explícita do conhecimento, além da ausência de representação interna simbólica do ambiente e memória das ações.

*Máquinas de estados finitos* constituem um método simplificado e prático para a construção de agentes reativos. Tais máquinas são inerentemente modulares, permitindo a construção de uma biblioteca de comportamentos primários juntamente com transições, baseadas em combinações condicionais. O comportamento pode então ser ligado pelas transições usando uma representação gráfica, como um autômato [10]. Porém, apesar da simplicidade conceitual dos métodos baseados em máquinas de estados finitos, a representação interna pode tornar-se inviável com o tempo e de acordo com a quantidade das informações processadas. Em média, a manutenção atinge  $\mathcal{O}(n^2)$  transições entre estados.

Para a implementação de um programa de agente reativo, uma abordagem geral e flexível é, em um primeiro momento, construir um interpretador de regras  $\langle \text{condição} \mapsto \text{ação} \rangle$ . Em

seguida, cria-se um conjunto de regras para ambientes de tarefas específicas.

### III. AGENTES REATIVOS ADAPTATIVOS

Esta seção apresenta a proposta de agentes adaptativos, em particular, na forma reativa. São introduzidas as definições de dispositivos adaptativos dirigidos por regras, bem como a organização de um agente adaptativo.

#### A. Dispositivos adaptativos

Um dispositivo é dito adaptativo quando este possui a capacidade de modificação de seu próprio comportamento, sem a interferência de agentes externos [11], [12]. Tal modificação espontânea ocorre em resposta ao histórico de operação e aos dados de entrada, resultando em comportamentos distintos ao longo do tempo [13]. A adaptatividade é adicionada como uma extensão aos formalismos já consolidados, aumentando seu poder de expressão [14], [15]. Os dispositivos adaptativos são definidos formalmente a seguir.

**Definição 1** (dispositivo guiado por regras). Um dispositivo guiado por regras é definido como  $ND = (C, NR, S, c_0, A, NA)$ , tal que  $ND$  é um dispositivo guiado por regras,  $C$  é o conjunto de todas as possíveis configurações,  $c_0 \in C$  é a configuração inicial,  $S$  é o conjunto de todos os possíveis eventos que são estímulos de entrada do dispositivo,  $\epsilon \in S$ ,  $A \subseteq C$  é o subconjunto de todas as configurações de aceitação (da mesma forma,  $F = C - A$  é o subconjunto das configurações de rejeição),  $NA$  é o conjunto de todos os possíveis símbolos de saída de  $ND$  como efeito da aplicação das regras do dispositivo,  $\epsilon \in NA$ , e  $NR$  é o conjunto de regras que definem  $ND$  através de uma relação  $NR \subseteq C \times S \times C \times NA$ .  $\square$

**Definição 2** (regra do dispositivo). Uma regra  $r \in NR$  tem a forma  $r = (c_i, s, c_j, z)$ ,  $c_i, c_j \in C$ ,  $s \in S$  e  $z \in NA$ , indicando que, em resposta a um estímulo  $s$ ,  $r$  muda a configuração corrente  $c_i$  para  $c_j$ , consome  $s$  e gera  $z$  como saída [6]. Uma regra  $r = (c_i, s, c_j, z)$  é dita compatível com a configuração corrente  $c$  se, e somente se,  $c_i = c$  e  $s$  é vazio ou igual ao estímulo de entrada corrente; neste caso, a aplicação da regra  $r$  move o dispositivo para a configuração  $c_j$ , denotada por  $c_i \Rightarrow^s c_j$ , e adiciona  $z$  ao fluxo de saída.  $\square$

**Definição 3** (aceitação de um fluxo de estímulos de entrada por um dispositivo). Um fluxo de estímulos de entrada  $w = w_1 w_2 \dots w_n$ ,  $w_k \in S - \{\epsilon\}$ ,  $k = 1, \dots, n$ ,  $n \geq 0$ , é aceito por um dispositivo  $ND$  quando  $c_0 \Rightarrow^{w_1} c_1 \Rightarrow^{w_2} \dots \Rightarrow^{w_n} c$  (de modo resumido,  $c_0 \Rightarrow^w c$ ), e  $c \in A$ . Da mesma forma,  $w$  é rejeitado por  $ND$  quando  $c \in F$ . A linguagem descrita pelo dispositivo guiado por regras  $ND$  é representada pelo conjunto  $L(ND) = \{w \in S^* \mid c_0 \Rightarrow^w c, c \in A\}$ .  $\square$

**Definição 4** (dispositivo adaptativo guiado por regras). Um dispositivo guiado por regras  $AD = (ND_0, AM)$ , tal que  $ND_0$  é um dispositivo e  $AM$  é um mecanismo adaptativo, é considerado adaptativo sempre que, para todos os passos de operação  $k \geq 0$  ( $k$  é o valor de um contador embutido  $T$  iniciado em zero e que é incrementado de uma unidade toda

vez que uma ação adaptativa não nula é executada),  $AD$  segue o comportamento do dispositivo subjacente  $ND_k$  até que a execução de uma ação adaptativa não nula inicie o passo de operação  $k + 1$  através de mudanças no conjunto de regras; em outras palavras, a execução uma ação adaptativa não nula em um passo de operação  $k \geq 0$  faz o dispositivo adaptativo  $AD$  evoluir do dispositivo subjacente  $ND_k$  para  $ND_{k+1}$ .  $\square$

**Definição 5** (operação do dispositivo adaptativo). O dispositivo adaptativo  $AD$  inicia sua operação na configuração  $c_0$ , com o formato inicial definido por  $AD_0 = (C_0, AR_0, S, c_0, A, NA, BA, AA)$ . No passo  $k$ , um estímulo de entrada move  $AD$  para uma configuração seguinte e inicia seu passo de operação  $k + 1$  se, e somente se, uma ação não-adaptativa for executada; dessa forma, estando o dispositivo  $AD$  no passo  $k$ , com o formato  $AD_k = (C_k, AR_k, S, c_k, A, NA, BA, AA)$ , a execução de uma ação adaptativa não nula leva a  $AD_{k+1} = (C_{k+1}, AR_{k+1}, S, c_{k+1}, A, NA, BA, AA)$ , onde  $AD = (ND_0, AM)$  é um dispositivo adaptativo com um dispositivo subjacente inicial  $ND_0$  e um mecanismo adaptativo  $AM$ ,  $ND_k$  é o dispositivo subjacente de  $AD$  no passo de operação  $k$ ,  $NR_k$  é o conjunto de regras não adaptativas de  $ND_k$ ,  $C_k$  é o conjunto de todas as configurações possíveis para  $ND$  no passo de operação  $k$ ,  $c_k \in C_k$  é a configuração inicial no passo  $k$ ,  $S$  é o conjunto de todos os eventos possíveis que são estímulos de entrada para  $AD$ ,  $A \subseteq C$  é o subconjunto as configurações de aceitação (da mesma forma,  $F = C - A$  é o subconjunto de configurações de rejeição),  $BA$  e  $AA$  são conjuntos de ações adaptativas (ambos contendo a ação nula,  $\epsilon \in BA \cap AA$ ),  $NA$ , com  $\epsilon \in NA$ , é o conjunto de todos os possíveis símbolos de saída de  $AD$  como efeito da aplicação de regras do dispositivo,  $AR_k$  é o conjunto de regras adaptativas definido pela relação  $AR_k \subseteq BA \times C \times S \times C \times NA \times AA$ , com  $AR_0$  definindo o comportamento inicial de  $AD$ ,  $AR$  é o conjunto de todas as possíveis regras adaptativas para  $AD$ ,  $NR$  é o conjunto de todas as possíveis regras não-adaptativas subjacentes de  $AD$ , e  $AM$  é o mecanismo adaptativo,  $AM \subseteq BA \times NR \times AA$ , a ser aplicado em um passo de operação  $k$  para cada regra em  $NR_k \subseteq NR$ .  $\square$

**Definição 6** (regras adaptativas do dispositivo). Regras adaptativas  $ar \in AR_k$  são da forma  $ar = (ba, c_i, s, c_j, z, aa)$  indicando que, em resposta a um estímulo de entrada  $s \in S$ ,  $ar$  inicialmente executa a ação adaptativa anterior  $ba \in BA$ ; a execução de  $ba$  é cancelada se esta elimina  $ar$  do conjunto  $AR_k$ ; caso contrário, a regra não-adaptativa subjacente  $nr = (c_i, s, c_j, z)$ ,  $nr \in NR_k$  é aplicada e, finalmente, a ação adaptativa posterior  $aa \in AA$  é executada [6].  $\square$

**Definição 7** (função adaptativa). Ações adaptativas podem ser definidas em termos de abstrações chamadas funções adaptativas, de modo similar às chamadas de funções em linguagens de programação usuais [6]. A especificação de uma função adaptativa deve incluir os seguintes elementos: (a) um nome simbólico, (b) parâmetros formais que referenciarão valores passados como argumentos, (c) variáveis que conterão valores de uma aplicação de uma ação elementar de inspeção,

(d) geradores que referenciam valores novos a cada utilização, e (e) o corpo da função propriamente dita.  $\square$

**Definição 8** (ações adaptativas elementares). São definidos três tipos de ações adaptativas elementares que realizam testes nas regras ou modificam regras existentes, a saber:

- 1) *ação adaptativa elementar de inspeção*: a ação não modifica o conjunto de regras, mas permite a inspeção deste para a verificação de regras que obedeçam um determinado padrão.
- 2) *ação adaptativa elementar de remoção*: a ação remove regras que correspondem a um determinado padrão do conjunto corrente de regras.
- 3) *ação adaptativa elementar de inclusão*: a ação insere uma regra que corresponde a um determinado padrão no conjunto corrente de regras.

Tais ações adaptativas elementares podem ser utilizadas no corpo de uma função adaptativa, incluindo padrões de regras que utilizem parâmetros formais, variáveis e geradores disponíveis.  $\square$

Observe que o mecanismo adaptativo pode ser visto como uma simples extensão do dispositivo não-adaptativo subjacente, o que preserva a integridade e as propriedades deste [6].

#### B. Agentes adaptativos

Um agente adaptativo é um dispositivo que observa seu ambiente através de sensores e age no próprio ambiente através de atuadores, de acordo com uma função  $f$  que determina seu comportamento, podendo o conjunto de regras de  $f$  modificar-se espontaneamente ao longo do tempo, em resposta ao histórico de operação e às percepções. A função  $f$  do agente realiza o mapeamento de um conjunto de percepções em uma ação [1], tal que  $f: 2^P \mapsto AC$ , no qual  $P$  é o conjunto de percepções e  $AC$  é o conjunto de ações. A modificação do conjunto de regras da função do agente ao longo do tempo ocorre por meio do mecanismo adaptativo, disparada pela execução de regras adaptativas.

**Definição 9** (agente adaptativo). Um agente adaptativo é definido como  $AA = (A_0, AM)$ , tal que  $A_0$  é um agente e  $AM$  é um mecanismo adaptativo. Para todos os passos de operação  $k \geq 0$  ( $k$  é o valor de um contador embutido  $T$  iniciado em zero e que é incrementado de uma unidade toda vez que uma ação adaptativa não nula é executada),  $AA$  segue o comportamento do agente subjacente  $A_k$  até que a execução de uma ação adaptativa não nula inicie o passo de operação  $k + 1$  através de alterações no conjunto de regras da função  $f$  do agente. Se o dispositivo subjacente  $A_0$  possuir características adaptativas, o agente  $AA$  é classificado como adaptativo multinível.  $\square$

A Figura 2 apresenta uma possível organização de um agente adaptativo, estendida a partir do modelo proposto por Russell e Norvig [1]. Observe que o agente adaptativo é formado por um agente convencional acrescido de um mecanismo adaptativo.

De acordo com a Figura 2, o mecanismo adaptativo é representado na forma de um tratador de ações adaptativas.

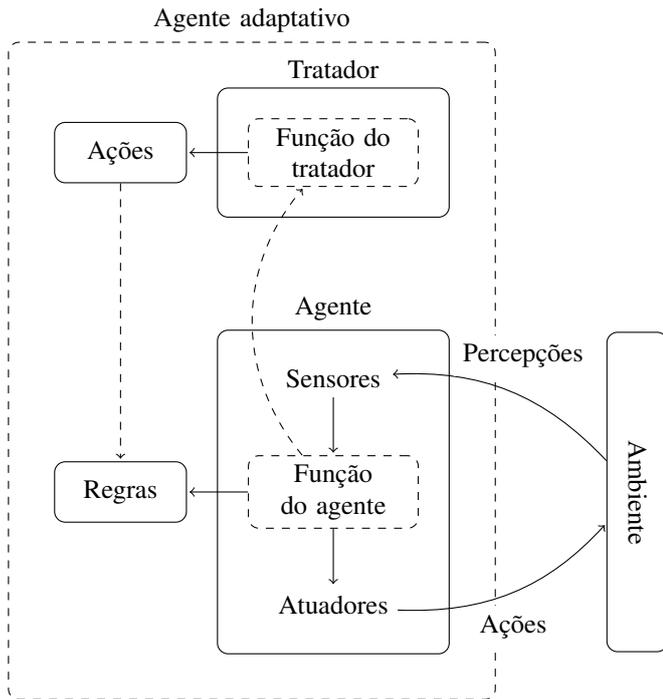


Figura 2. Organização de um agente adaptativo, estendida a partir do modelo proposto por Russell e Norvig [1].

Este intercepta pedidos de tratamento a partir da função do agente subjacente e realiza as modificações de acordo com padrões estabelecidos previamente em seu conjunto de ações. A modificação do conjunto de regras da função  $f$  do agente, através da execução de ações adaptativas, proporciona a evolução do agente subjacente no tempo, conforme ilustra a Figura 3.

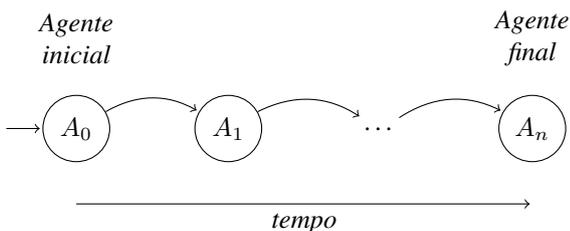


Figura 3. Evoluções sucessivas do agente subjacente no tempo, através da efetivação de ações adaptativas.

O conjunto de regras da função  $f$  do agente é acrescido de ações adaptativas anteriores e posteriores, conforme ilustra a Figura 4. Tais ações disparam pedidos de tratamento ao componente tratador do agente, que efetivamente realiza as modificações no conjunto de regras.

A Figura 4 apresenta partes hachuradas que indicam a existência de ações adaptativas associadas às regras correspondentes. No exemplo, a regra 1 possui apenas uma ação adaptativa anterior, a regra 2 não possui ações adaptativas associadas, e a regra  $n$  possui duas ações adaptativas, uma anterior e uma posterior.

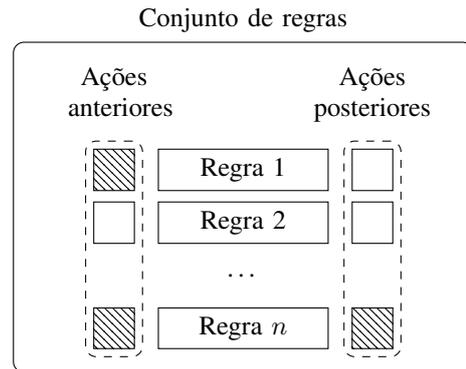


Figura 4. Formato de um conjunto de regras acrescido de ações adaptativas. As partes hachuradas indicam a existência de ações adaptativas associadas às regras correspondentes.

É possível utilizar adaptatividade multinível [16], [17], [18] na especificação de um agente, conforme ilustra a organização apresentada na Figura 5. Observe que um agente adaptativo de nível  $n$  possui  $n$  componentes tratadores, tal que um tratador  $k + 1$  intercepta pedidos de tratamento do nível  $k$ , hierarquicamente inferior, e realiza alterações no conjunto de regras ou ações disponível em tal nível. É importante notar que o agente encontra-se no nível 0 da organização do modelo.

A utilização de um agente adaptativo na resolução de um problema pode conferir expressividade ao modelo e simplificação da representação da lógica interna em relação ao seu correspondente não-adaptativo.

### C. Agentes reativos como dispositivos subjacentes

A função do agente pode ser implementada utilizando-se estratégias básicas de projeto que refletem quais informações serão utilizadas no processo de tomada de decisão. O tipo mais simples é o agente reativo, que atua somente sobre a percepção corrente, ignorando o histórico de percepções recebidas; a tomada de decisão baseia-se estritamente em um conjunto de regras condicionais [1].

**Definição 10** (agente reativo). Um agente reativo é definido como  $RA = (Q, P, AC, \Gamma, \delta)$ , tal que  $Q$  é o conjunto de estados internos do agente,  $P$  é o conjunto de percepções,  $AC$  é o conjunto de ações,  $\Gamma$  é o mapeamento,  $\Gamma: Q \times 2^P \mapsto Q$ , e  $\delta$  é uma função que mapeia regras em ações,  $\delta: P \mapsto AC$ .  $\square$

Agentes reativos possuem uma característica limitante: a tomada de decisão atua apenas sobre a percepção corrente, implicando na necessidade da observação total do ambiente; caso contrário, o agente potencialmente falhará [1]. A inclusão de uma camada adaptativa em um agente reativo permite tratar situações previstas, mas não esperadas no contexto corrente; através da execução de ações adaptativas, o agente tem seu conjunto de regras modificado, acomodando, portanto, um novo contexto a partir da observação do ambiente. A Figura 6 ilustra as modificações do conjunto de regras da função do agente reativo adaptativo ao longo do tempo para acomodar novos contextos.

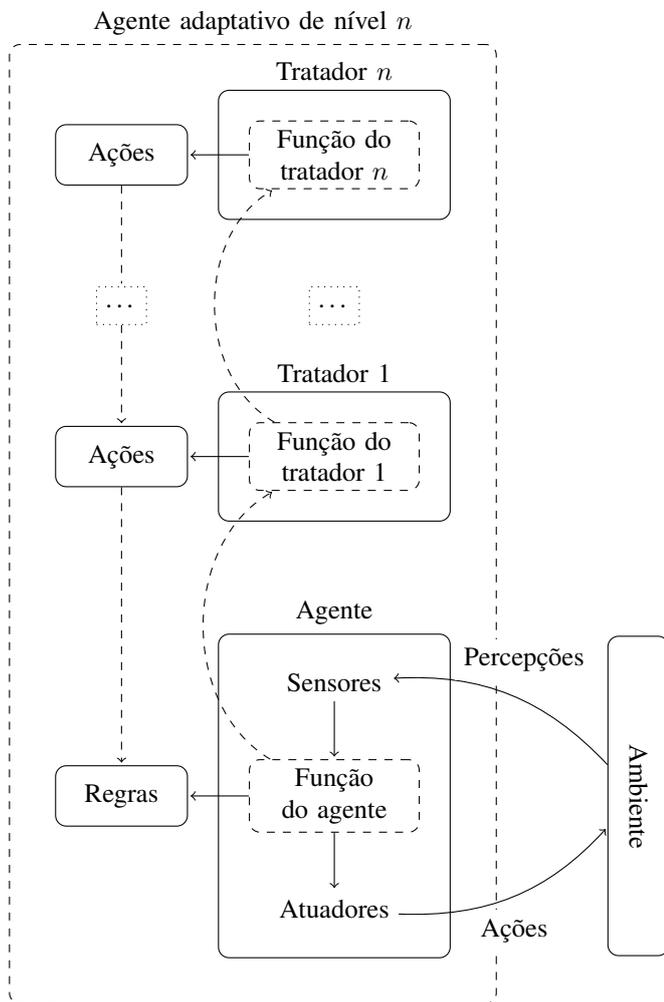


Figura 5. Organização de um agente adaptativo multinível, estendida a partir do modelo apresentado na Figura 2.

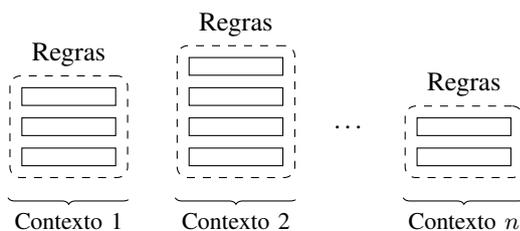


Figura 6. Modificações do conjunto de regras da função do agente reativo adaptativo ao longo do tempo para acomodar novos contextos.

Novos contextos proporcionam resolução incremental, através da divisão de um problema em subproblemas menores [19], [20]. A resolução de um problema em um espaço de abstração mais simples pode ser utilizada para o tratamento de outros problemas em níveis de abstrações superiores [21]. O processo é então repetido até que o problema original seja resolvido em seu espaço original de abstração [22]. Tal estratégia de resolução pode produzir reduções significativas no espaço de busca [23], [24], [25].

Observe que, dado um agente reativo simples, é possível adicionar uma camada adaptativa de tal forma que este continue reativo em sua estrutura, mas agindo por contexto na execução.

#### IV. EXPERIMENTOS

Esta seção apresenta um estudo de caso para avaliação de um agente adaptativo reativo utilizando o tradicional problema do mundo de wumpus, uma variante de um jogo de computador [26] apresentada por Genesereth [1] como ambiente de testes para técnicas em inteligência artificial. O mundo de wumpus consiste em uma grade bidimensional contendo um número determinado de buracos, um monstro e um agente.

No mundo de wumpus, o agente inicia sua iteração na posição (1,1) da grade e sua tarefa consiste em evitar o monstro e os buracos, encontrar ouro e sair do ambiente pela mesma posição a qual iniciou. O agente pode perceber uma brisa em posições adjacentes aos buracos, um fedor em posições adjacentes ao monstro, e um brilho em posições adjacentes ao ouro. Existem variações do mundo de wumpus que permitem ao agente armar-se com uma ou mais flechas, podendo atirar no monstro quando encontrá-lo [27].

Em geral, os ambientes do mundo de wumpus permitem que o agente recupere o ouro e saia em segurança. Entretanto, existem situações em que o ouro estará inalcançável (por exemplo, dentro de um buraco) ou que o agente deverá enfrentar o monstro, podendo morrer no confronto [27], [1].

##### A. Especificação do agente adaptativo reativo

O agente adaptativo reativo proposto para o mundo de wumpus foi especificado de acordo com as seguintes regras iniciais:

- O agente pode locomover-se pelo ambiente, evitando buracos e o monstro, de acordo com a observação das percepções correntes.
- Ao perceber a existência do monstro em uma posição adjacente, o agente poderá optar por disparar uma flecha, na tentativa de matá-lo.
- Na percepção de um brilho em uma posição adjacente, o agente tentará recuperar o ouro.

O contexto inicial do agente prevê que este locomova-se e sobreviva no ambiente, e procure pelo ouro. Adicionalmente, as seguintes regras disparam ações adaptativas:

- Caso o agente opte por disparar uma flecha, na tentativa de matar o monstro, e este disparo for bem-sucedido, haverá uma percepção de um grito no ambiente, indicando que o monstro está morto. Neste caso, na percepção de um grito, uma ação adaptativa removerá a regra que trata da percepção de fedor.

- Caso o agente recupere o ouro, através da percepção de brilho, uma ação adaptativa removerá a própria regra que determina a recuperação do ouro e adicionará uma regra que determina a percepção da entrada do ambiente, permitindo que o agente saia pela mesma posição que entrou.

Observe que, ao matar o monstro, não existe mais a necessidade do agente em possuir a regra referente ao confronto (a menos, é claro, que o agente esteja atuando em uma variação do mundo de wumpus que admite a existência de mais monstros). Da mesma forma, não existe a necessidade de possuir a regra referente à percepção do ouro, dado que este já esteja recuperado pelo agente. A Figura 7 ilustra os possíveis contextos tratados pelo agente adaptativo reativo, de acordo com a execução das ações adaptativas.

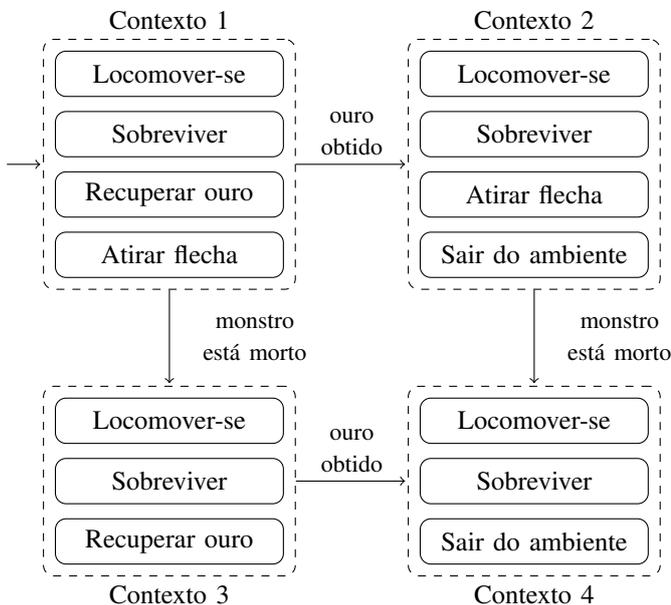


Figura 7. Possíveis contextos tratados pelo agente adaptativo reativo para o problema do mundo de wumpus.

É importante destacar que o conjunto de regras do agente adaptativo reativo representa o conhecimento necessário e suficiente para tratamento do contexto corrente. As ações adaptativas permitem que o conjunto de regras modifique-se ao longo do tempo para acomodar situações previstas, mas não esperadas.

### B. Simulação do ambiente

A interação do agente adaptativo reativo no mundo de wumpus foi simulada utilizando-se um software de especificação e simulação de agentes desenvolvido por Jill Zimmerman<sup>1</sup> (Goucher College, Estados Unidos). Foram gerados 14 ambientes de dimensão  $4 \times 4$ , com diferentes quantidades e localizações de obstáculos, conforme ilustra a Figura 8.

O software de simulação utilizado para este experimento apresenta as seguintes medidas de desempenho da interação do

agente com o ambiente [1], permitindo o cálculo da pontuação total ao final da simulação:

- 1) Bônus de 1000 pontos ao recuperar o ouro (tal ação é uma parte importante da tarefa do agente no ambiente).
- 2) Penalização de  $-1000$  pontos quando o agente morre no ambiente. Esta situação ocorre quando o agente entra em uma posição contendo um buraco ou um monstro (ainda vivo).
- 3) Penalização de  $-1$  ponto para cada ação tomada pelo agente no ambiente (mesmo que o resultado da ação seja um bônus).
- 4) Penalização de  $-10$  pontos por utilizar uma flecha, na tentativa de matar o monstro.

O experimento consistiu na simulação da interação de quatro tipos de agentes – reativo simples, baseado em modelos, baseado em objetivos e adaptativo reativo – nos ambientes ilustrados na Figura 8, avaliando seus desempenhos globais. A simulação foi repetida 10 vezes para cada par ordenado  $\langle agente, ambiente \rangle$ , totalizando 560 execuções. Os resultados obtidos são exibidos na Figura 9.

De acordo com a Figura 9, é possível observar semelhanças na execução dos agentes em um mesmo ambiente. De modo particular, os agentes baseados em modelos e objetivos apresentaram execuções significativamente próximas. O agente adaptativo reativo apresentou uma execução relativamente melhor em relação ao agente reativo simples e, em alguns ambientes, teve pontuação próxima a dos agentes baseados em modelos e objetivos. Como melhoria, um refinamento das regras que determinam o trajeto a ser percorrido pode contribuir com a redução do número de passos do agente no ambiente e aumentar suas chances de sobrevivência.

## V. CONSIDERAÇÕES FINAIS

Este artigo apresentou uma proposta de organização de agentes adaptativos como uma alternativa viável às iniciativas existentes. A utilização de novos contextos através do fenômeno da adaptatividade apresenta benefícios importantes, tais como:

- 1) divisão de trabalho, permitindo que cada contexto tenha um comportamento específico dado um subproblema, reduzindo o espaço de regras para um subconjunto necessário e suficiente.
- 2) abstração, proporcionando representações distintas de acordo com o nível do subproblema tratado, sem entrar no mérito da representação do problema original.
- 3) reuso, de tal forma que contextos possam ser reaproveitados parcial ou totalmente, conforme a similaridade entre subproblemas.
- 4) facilidade de manutenção, na qual um contexto pode ter sua lógica interna atualizada sem comprometer os demais (desde que a integridade da resolução do subproblema seja preservada).

Agentes adaptativos apresentam conveniências na especificação de um conjunto de regras, organizando-as por contexto. Tal característica permite a simplificação do problema original, dividindo-o em subproblemas menores. Adicionalmente, a execução torna-se mais eficiente, dado que um contexto só

<sup>1</sup>Disponível em <http://phoenix.goucher.edu/~jilllz/cs340/>.

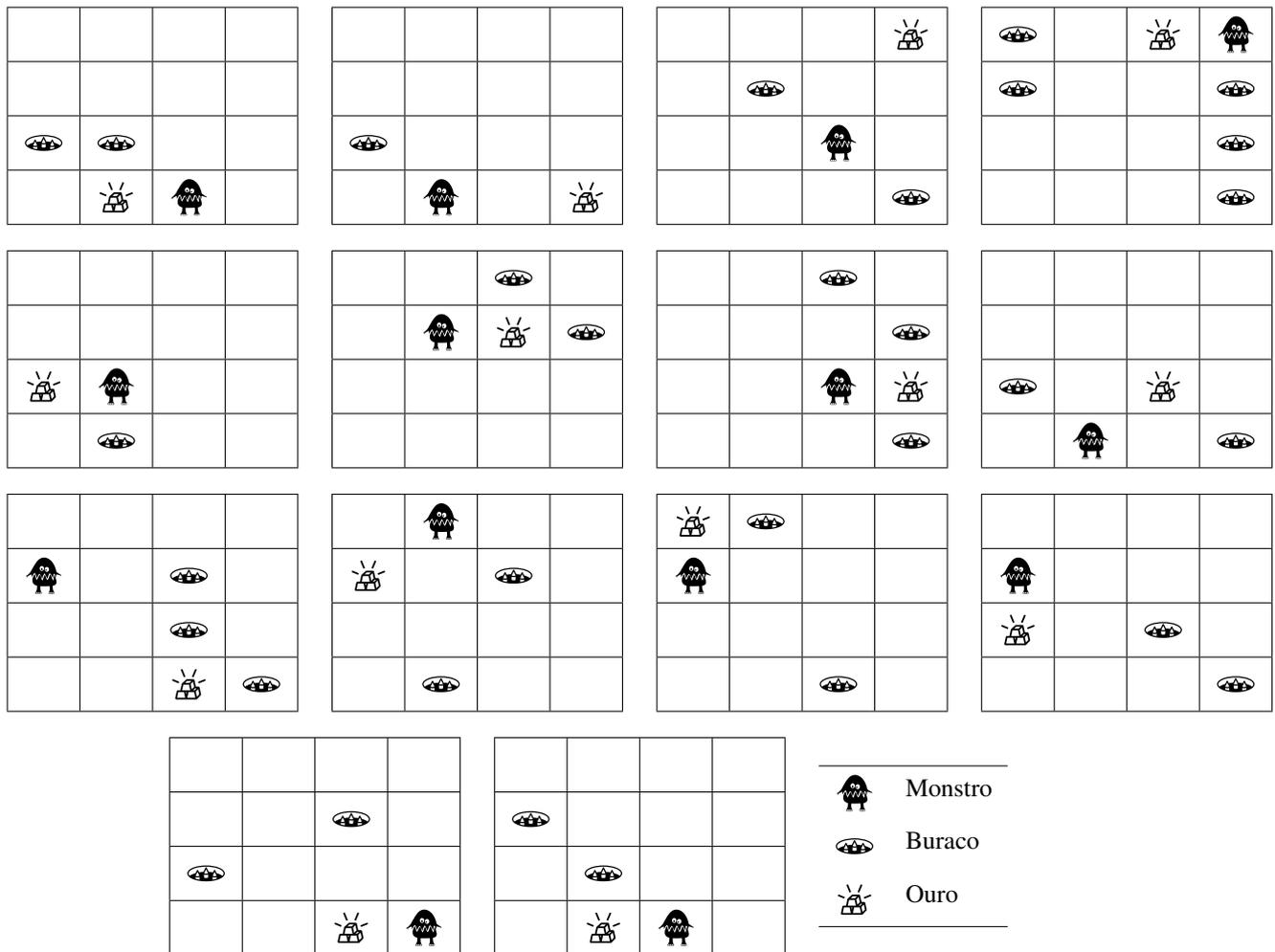


Figura 8. Ambientes gerados com diferentes quantidades e localizações de obstáculos para simulação da interação do agente adaptativo reativo no mundo de wumpus.

será ativado se efetivamente for necessário. Outras técnicas existentes podem ser combinadas [28], de forma a tratar especificamente cada subproblema.

#### REFERÊNCIAS

- [1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003.
- [2] G. Booch, *Object-Oriented Analysis and Design with Applications*, 2nd ed., D. J. et al., Ed. Addison Wesley Longman, Inc, 1994.
- [3] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, p. 31–40, 1994.
- [4] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, pp. 7–38, 1998.
- [5] A. Ahmad, M. S. Ahmad, and M. Z. Yusof, "An exploratory review of software agents," *International Symposium on Information Technology*, 2008.
- [6] J. José Neto, "Adaptive rule-driven devices: general formulation and case study," in *International Conference on Implementation and Application of Automata*, 2001.
- [7] T. Pedrazzi, A. H. Tchembra, and R. L. A. Rocha, "Adaptive decision tables – a case study of their application to decision-taking problems," in *Proceedings of International Conference on Adaptive and Natural Computer Algorithms – ICANNGA 2005*, 2005.
- [8] H. Pistori and J. J. Neto, "Adaptree – proposta de um algoritmo para indução de Árvores de decisão baseado em técnicas adaptativas," in *Anais da Conferência Latinoamericana de Informática – CLEI 2002*, Montevideo, Uruguai, Novembro 2002.
- [9] M. C. and J. José Neto, "Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos," in *V PROPOR – Encontro para o processamento computacional de Português falado e escrito*, 2000.
- [10] G. M. Heckel, F. W. P.; Youngblood and N. S. Ketkar, "Representational complexity of reactive agents," in *IEEE Symposium on Computational Intelligence and Games (CIG)*, 2010, p. 257–264.
- [11] J. José Neto, "Contribuições à metodologia de construção de compiladores," Thesis (Livre-docência), Escola Politécnica, Universidade de São Paulo, 1993.
- [12] —, "Adaptive automata for context-dependent languages," *SIGPLAN Notices*, vol. 29, no. 9, pp. 115–124, 1994.
- [13] —, "Um levantamento da evolução da adaptatividade e da tecnologia adaptativa," *IEEE Latin America Transactions*, vol. 5, pp. 496–505, 2007.
- [14] H. Pistori, "Tecnologia em engenharia de computação: estado da arte e aplicações," PhD thesis, Escola Politécnica, Universidade de São Paulo, 2003.
- [15] P. R. M. Cereda and J. José Neto, "Utilizando linguagens de programação orientadas a objetos para codificar programas adaptativos," in *Memórias do IX Workshop de Tecnologia Adaptativa – WTA 2015*, 2015, pp. 2–9.
- [16] R. I. Silva Filho, R. L. A. Rocha, and R. H. G. Guiraldelli, *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence: Papers from the Ray Solomonoff 85th Memorial Conference, Melbourne, VIC, Australia, November 30 – December 2, 2011*. Springer Berlin Heidelberg, 2013, ch. Learning in the Limit: A Mutational and Adaptive Approach, pp. 106–118.
- [17] R. I. Silva Filho and R. L. A. Rocha, *Adaptive and Natural Computing Algorithms: 10th International Conference, ICANNGA 2011, Ljubljana*,

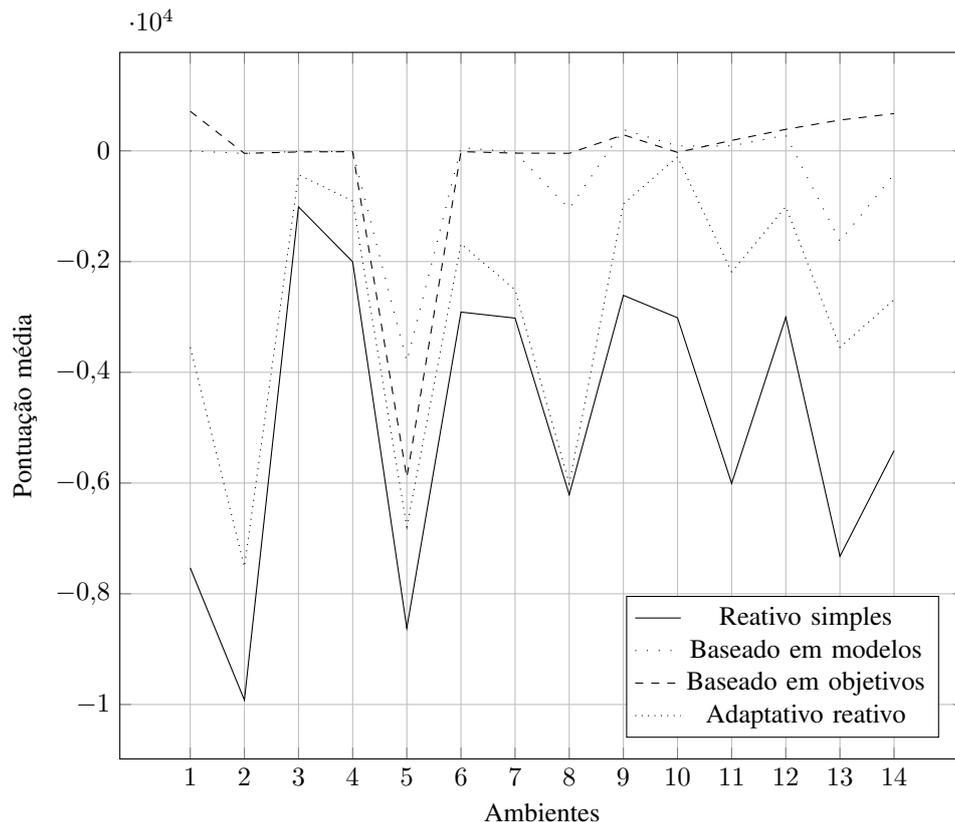


Figura 9. Pontuação média dos quatro agentes (reativo simples, baseado em modelos, baseado em objetivos e adaptativo reativo) em 14 ambientes do mundo de wumpus.

- Slovenia, April 14-16, 2011, *Proceedings, Part II*. Springer Berlin Heidelberg, 2011, ch. Adaptive Finite Automaton: A New Algebraic Approach, pp. 275–284.
- [18] R. I. Silva Filho, “Uma nova formulação algébrica para o autômato finito adaptativo de segunda ordem aplicada a um modelo de inferência indutiva,” Tese de doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 2011.
- [19] C. Knoblock, *Generating Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning*. Springer US, 1993.
- [20] Y. Anzai and H. A. Simon, “The theory of learning by doing,” *Psychological Review*, no. 86, pp. 124–140, 1979.
- [21] C. Knoblock, “Learning abstraction hierarchies for problem solving,” in *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1990, pp. 923–928.
- [22] G. Pólya, *How to Solve It*. Princeton University Press, 1945.
- [23] M. Minsky, *Computers and Thought*. New York, NY: McGraw-Hill, 1963, ch. Steps toward artificial intelligence, pp. 406–450.
- [24] A. Newell and H. A. Simon, *Contemporary Approach to Creative Thinking*. Atherton Press, 1962, ch. The processes of creative thinking, pp. 63–119.
- [25] C. Knoblock, “Search reduction in hierarchical problem solving,” in *AAAI’91 Proceedings*, 1991, pp. 686–691.
- [26] G. Yob, “Hunt the wumpus,” *People’s Computer Company*, vol. 2, no. 1, 1973.
- [27] G. M. Khan, J. F. Miller, and D. M. Halliday, *Advances in modeling adaptive and cognitive systems*. UEFS, 2010, ch. Intelligent agents capable of developing memory of their environment, pp. 77–114.
- [28] P. R. M. Cereda and J. José Neto, “Adaptive data mining: Preliminary studies,” *IEEE Latin America Transactions*, vol. 12, no. 7, pp. 1258–1270, October 2014.



**Renata Luiza Stange** atualmente é doutoranda do Programa de Pós-Graduação em Engenharia de Computação do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, atuando como pesquisadora no Laboratório de Linguagens e Técnicas Adaptativas do PCS. Mestre em Ciências, também pelo Programa de Pós-Graduação em Engenharia de Computação do PCS/EPUSP (2011). Bacharel em Análise de Sistemas pela Universidade Estadual do Centro-Oeste (2002). Atua como docente na Universidade Tecnológica Federal do Paraná (UTFPR). Tem experiência na área de Ciência da Computação, particularmente em dispositivos adaptativos, tecnologia adaptativa e suas aplicações à inteligência artificial, aprendizagem de máquina e reconhecimento de padrões.



**Paulo Roberto Massa Cereda** é graduado em Ciência da Computação pelo Centro Universitário Central Paulista (2005) e mestre em Ciência da Computação pela Universidade Federal de São Carlos (2008). Atualmente, é doutorando do Programa de Pós-Graduação em Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, atuando como aluno pesquisador no Laboratório de Linguagens e Técnicas Adaptativas do PCS. Tem experiência na área de Ciência da Computação, com ênfase em Teoria da Computação, atuando principalmente nos seguintes temas: tecnologia adaptativa, autômatos adaptativos, dispositivos adaptativos, linguagens de programação e construção de compiladores.



**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 Técnicas Adaptativas 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.