

**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA POLITÉCNICA**

**MIGUEL ANGELO DE ABREU DE SOUSA**

**Mapeamento de Ambientes Desconhecidos  
por Robôs Móveis Utilizando  
Autômatos Adaptativos**

São Paulo  
2006

**Miguel Angelo de Abreu de Sousa**

**Mapeamento de Ambientes  
Desconhecidos por Robôs Móveis  
Utilizando Autômatos Adaptativos**

Dissertação apresentada à Escola Politécnica  
da Universidade de São Paulo para obtenção  
do título de Mestre em Engenharia Elétrica.

Área de Concentração: Engenharia da  
Computação e Sistemas Digitais

Orientador: Prof. Dr. André Riyuiti Hirakawa

São Paulo  
2006

A Greice, Alcides, Carmem e Marta.

## **AGRADECIMENTOS**

Ao Professor André Riyuiti Hirakawa, pela valorosa orientação, constante disponibilidade e paciência nos momentos de turbulência.

Ao Professor João José Neto, pela idéia inicial que permitiu que este e outros projetos fossem desenvolvidos e pelas inúmeras conversas que me ajudaram num âmbito muito maior do que a realização do presente trabalho.

Ao CNPq pelo suporte à pesquisa realizada (processo:133067/2004-1).

À minha irmã, pelos diversos auxílios e consultas técnicas que viabilizaram a conclusão do trabalho.

Aos meus pais, pela formação básica e fundamental que me concederam.

E à minha esposa, por incentivar e apoiar de forma incondicional esta nova jornada.

## RESUMO

Sousa, M. A. A. **Mapeamento de Ambientes Desconhecidos por Robôs Móveis Utilizando Autômatos Adaptativos**. 2005. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

Este trabalho apresenta uma alternativa aos métodos clássicos empregados na construção autônoma de um mapa do ambiente no qual um robô móvel está inserido – procedimento chamado de mapeamento robótico. Para tal elaborou-se a proposta, implementação e avaliação de uma estrutura para o registro dos dados coletados sobre a topologia do local pelo conjunto de sensores do robô, empregando-se um formalismo adaptativo: os Autômatos Adaptativos. Aliado a esta arquitetura, propõe-se ainda um algoritmo que permita ao robô móvel explorar completamente o espaço livre e alcançável de um ambiente desconhecido com a finalidade de adquirir informações sobre a configuração do local durante o seu deslocamento. Este algoritmo é igualmente implementado empregando-se os Autômatos Adaptativos e seu comportamento também é avaliado no presente trabalho.

## ABSTRACT

Sousa, M. A. A. **Mapping Unknown Environments by Mobile Robots Using Adaptive Automata**. 2005. Dissertation (Mastership) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

The present work shows an alternative to the classic methods applied to the autonomous map building of an unknown environment in which a mobile robot is placed. This procedure is called robotic mapping. In order to achieve this purpose, it has been proposed, implemented and evaluated a framework to register the collected data from the environment by the robots sensors set, employing an adaptive formalism: Adaptive Automata. Also, it has been proposed an algorithm that allows the mobile robot to fully explore any reachable free space around. So, it can acquire information on the environment configuration during its displacement on the place. Such algorithm is also implemented using Adaptive Automata and its behavior is evaluated in this work too.

## 1. INTRODUÇÃO

A área de pesquisa de robótica móvel se diferencia da robótica tradicional por equipar os agentes robóticos com algum sistema de deslocamento a fim de permitir que estes sejam capazes de se movimentar no ambiente no qual estão inseridos. Tal capacidade de locomoção é adicionada às habilidades clássicas de percepção do espaço físico ao redor, através de conjuntos de sensores, e atuação neste mesmo espaço, através de conjuntos de atuadores, de modo a possibilitar que os robôs desempenhem tarefas complexas.

Entre as possíveis aplicações da robótica móvel, destacam-se, além de atuações nas áreas industriais e residenciais, as operações de resgate em situações de catástrofe, exploração marítima e planetária e a execução de tarefas agrícolas. Entretanto, apesar de ser estudada há três décadas, esta área de pesquisa apresenta ainda muitos desafios aos pesquisadores de robótica e inteligência artificial.

Brooks (1991), ao comparar a evolução da capacidade humana de percepção e mobilidade em ambientes dinâmicos, argumenta que estas habilidades, consideradas triviais, demoraram muito mais tempo para serem aperfeiçoadas do que as habilidades comumente consideradas como demonstração de inteligência, e que tais competências são essenciais para o desenvolvimento de agentes robóticos autônomos.

Entre outras habilidades, Costa (2003) expõe que robôs autônomos devem ser capazes de adaptar-se aos ambientes de operação, aprender conforme adquirem mais experiência e realizar escolhas frente às mudanças apresentadas por estes ambientes, pelo refinamento de suas ações.

Tendo tal direção, o presente trabalho apresenta o desenvolvimento do processo de mapeamento robótico, tarefa considerada fundamental para a robótica móvel, através do emprego de um formalismo de computação adaptativo.

## 1.1 Objetivo

O objetivo deste trabalho é o estudo de uma arquitetura de registro de dados que possibilite a construção de representações abstratas de ambientes desconhecidos por robôs móveis. A arquitetura proposta deve permitir sua posterior utilização na navegação do robô por este ambiente.

Pretende-se que a arquitetura apresentada seja uma alternativa às abordagens clássicas do assunto e, para alcançar tal objetivo, emprega-se um formalismo adaptativo como estrutura de registro e tratamento das informações adquiridas pelo conjunto de sensores de um robô móvel. Desta forma, excluem-se do contexto do presente trabalho os sistemas de sensores e motores que compõem o agente robótico.

Com o propósito de possibilitar ao robô móvel mapear completamente o ambiente em que está inserido, propõe-se ainda um algoritmo que permita a tal robô explorar inteiramente este ambiente, de modo que a construção da representação abstrata do espaço físico seja realizada durante o seu deslocamento por este local.

## 1.2 Justificativas

As primeiras abordagens da questão da mobilidade de robôs que aparecem na literatura apresentam, como motivação, a necessidade de uma maior automação dos processos industriais, como por exemplo, o transporte de peças entre diferentes estações e a necessidade de se ter um mesmo robô desempenhando suas tarefas em locais distintos (DRAKE; MCVEY; IÑIGO, 1987). Estas abordagens iniciais empregavam linhas pintadas no chão ou fios

enterrados sob o piso para guiar robôs em seu deslocamento entre duas estações de trabalho (TSUMURA, 1986; DRAKE; MCVEY; IÑIGO, 1988).

Posteriormente, com o aumento da necessidade da ampliação da mobilidade e da área de deslocamento dos robôs, inicia-se a utilização de mapas da área de trabalho manualmente construídos e manualmente registrados na memória do agente robótico (FOK; KABUKA, 1991). Tal abordagem apresenta três principais problemas:

1. Complexidade computacional. O registro de uma representação geométrica completa do ambiente onde um robô irá trabalhar aumenta a complexidade computacional dos processos subseqüentes de localização e planejamento de trajetória. Este problema torna a solução proibitiva para aplicações reais (ZIMMER, 2000).
2. Não automatização do processo de construção de mapas. Cada ambiente onde o agente robótico opera deve de ser previamente mapeado e, este mapa, manualmente registrado em sua memória. Devido ao caráter não automático deste processo, tal problema inviabiliza a solução para aplicações em larga escala (BROOKS, 1985).
3. Possibilidade de robôs operando em áreas de risco ou perigosas aos seres humanos. Mineração, operações em áreas de desastre, explorações marítimas, planetárias e espaciais são exemplos de situações nas quais os agentes robóticos devem ser capazes de construir autonomamente a representação do ambiente para que possam desempenhar suas tarefas adequadamente (KATO; MURAKAMI; OHNISHI, 2002).

Todas estas questões apontam para uma solução automática, na qual o robô, quando presente em um ambiente desconhecido, seja capaz de obter informações sobre a configuração deste ambiente através de seu conjunto de sensores e memorizar estas informações em uma estrutura que permita uma posterior utilização no processo de navegação (GÜRTLER; VAJTA; NAGY, 1997), este procedimento é chamado de mapeamento robótico (em inglês tal procedimento é chamado de “robotic mapping” ou “robotic map making”).

A área de pesquisa de mapeamento robótico pode ser dividida em dois ramos distintos: a abordagem métrica e a abordagem topológica.

Mapas métricos representam as propriedades geométricas do ambiente. Tais mapas são, geralmente, mais fáceis de serem construídos e seu emprego obtém uma representação detalhada do ambiente, entretanto, requerem, normalmente, grande quantidade de memória para registro de suas informações e exigem alto poder computacional de seus sistemas de planejamento de trajetória. As técnicas empregadas nas pesquisas de Movarec e Elfes (1985), Yamauchi (1997), Marco, Prattichizzo e Vicino (2000) e Wolf e Sukhatme (2003) são exemplos da utilização de mapas métricos.

Mapas topológicos descrevem a interligação entres pontos diferentes do espaço, representando o mundo como locais importantes e os caminhos que os conectam. Técnicas topológicas requerem menor espaço de memória e menor poder computacional, entretanto, os modelos construídos por tais técnicas não representam tão detalhadamente o mundo e a definição dos locais importantes ainda é uma questão em estudo. As técnicas empregadas em nas pesquisas de Brooks (1985), Innocenti (1994) e Jennings, Watts e Tanis (1998) são exemplos da utilização de mapas topológicos.

A proposta apresentada neste trabalho utiliza um formalismo adaptativo chamado Autômatos Adaptativos, proposto por Neto (1993), os quais possuem a característica de alterar sua configuração interna – e, portanto, seu comportamento – frente a estímulos externos.

O formalismo adaptativo foi escolhido como estrutura de representação de ambientes devido à sua capacidade inerente de utilização progressiva da memória do sistema, ou seja, a memória utilizada no processo é consumida de acordo com a área já mapeada. Além disso, o trabalho de Neto (1994) demonstra que os Autômatos Adaptativos possuem poder computacional equivalente à Máquina de Turing, o que permite que tal formalismo possa representar uma ferramenta confiável para modelagem de ambientes físicos.

O presente trabalho emprega também Autômatos Adaptativos como estrutura de decisão no processamento das informações encaminhadas pelo conjunto de sensores durante os deslocamentos de exploração pelo ambiente. Novamente, a capacidade adaptativa da ferramenta empregada possibilita o tratamento de informações similares e o desempenho de comportamentos distintos de acordo com os diferentes estágios da exploração do local em questão.

Desta forma, propõe-se que a estrutura apresentada figure como uma alternativa às clássicas abordagens do tema, na qual seja possível inferir no mapa construído características métricas das configurações do ambiente explorado e representar, de forma elementar, conexões entre espaços distintos e posições especiais neste ambiente.

### **1.3 Conteúdo e Organização**

Com o objetivo de apresentar a pesquisa desenvolvida, o restante deste trabalho está estruturado de acordo com a seguinte organização: O capítulo 2 descreve a área de pesquisa do mapeamento robótico e exploração de ambientes desconhecidos através da apresentação de trabalhos relacionados à robótica móvel e o capítulo 3 descreve o modelo da nossa proposta de execução destes processos de mapeamento e exploração. O capítulo 4 apresenta a Teoria dos Autômatos Adaptativos e os capítulos 5 e 6 detalham a configuração e operação dos autômatos responsáveis pelo registro de informações sobre o ambiente e pela definição dos movimentos que compõem a trajetória de exploração deste ambiente, respectivamente. Finalmente, o capítulo 7 apresenta as simulações do modelo proposto e a análise dos resultados do comportamento deste modelo, o capítulo 8 apresenta as considerações finais do presente trabalho e suas perspectivas de continuidade são apresentadas no capítulo 9.

## **2. PESQUISAS NA ÁREA DE MAPEAMENTO E EXPLORAÇÃO DE AMBIENTES**

Este capítulo descreve a área de pesquisa da robótica móvel aplicada ao processo de mapeamento robótico e ao seu procedimento complementar de exploração de ambientes desconhecidos, pesquisa que se apresentou como essencial para o desenvolvimento do presente trabalho.

### **2.1 Utilização de autômatos na robótica móvel**

Existem diversas pesquisas que empregam máquinas de estados e autômatos – acrescidos de outras funcionalidades ou não – em aplicações de robótica móvel. Fok e Kabuka (1991), por exemplo, utilizam uma máquina de estados finita para o controle dos comportamentos exibidos pelo robô durante a navegação por um ambiente já mapeado. Tais comportamentos podem ser, por exemplo, prosseguir em curso, chegada a um local demarcado, partida de um local demarcado, encontro de obstáculo que impede o prosseguimento da movimentação, posição final alcançada, chegada a um local demarcado que não possui conexão com a posição final, e mudanças internas de estado ocorrem de acordo com informações externas coletadas pelos sensores e com a sua representação interna do ambiente.

Brooks (1986) apresenta, em um trabalho clássico, a arquitetura de subordinação (“subsumption architecture”) para o controle do sistema de deslocamento de robôs móveis. A arquitetura é composta por uma sobreposição de camadas que são responsáveis por diferentes comportamentos simples exibidos pelo robô durante o processo de navegação. As camadas são implementadas em máquinas de estado finitas aumentadas, ou seja, que possuem atributos como supressores, memória e inibidores adicionados à estrutura da máquina de estados. Cada

camada possui um processamento assíncrono e independente de outras camadas, as camadas comunicam-se entre si e camadas superiores podem suprimir as saídas de camada inferiores.

O trabalho de Aoki et al. (1996) emprega um formalismo modificado com base nos autômatos finitos, chamado “learning automata”. Tal formalismo permite o aprendizado e evolução de comportamento do sistema de navegação de um robô móvel que possui um mapa previamente construído do ambiente com a finalidade de guiá-lo desde uma posição inicial até uma posição final, evitando a colisão com obstáculos fixos ou móveis.

Destaca-se também o trabalho de Prado e Abe (2003), que emprega autômatos paraconsistentes no controle do deslocamento de robôs. Autômatos Paraconsistentes fundamentam-se na alteração das características dos autômatos finitos de forma a poder tratar informações inconsistentes. Neste trabalho, Autômatos Paraconsistentes são empregados com a finalidade de conduzir um robô pelo ambiente desde uma posição inicial até uma posição final, lidando com as informações, às vezes conflitantes, vindas do conjunto de sensores do robô.

## **2.2 Mapeamento de ambientes**

A técnica de representações de espaços físicos utilizando registro de informações métricas sobre o ambiente tem como um dos principais trabalhos a pesquisa de Movarec e Elfes (1985), que introduz esta representação em uma estrutura de dados chamada de Grades de Evidência, isto é, um conjunto de células disposta no plano cartesiano, no qual cada célula recebe um valor proporcional à ocupação daquela região. As células podem ser classificadas em livres, ocupadas e desconhecidas de acordo com as informações coletadas pelo conjunto de sonares que o robô utiliza. Os autores argumentam que a utilização de sonares é uma alternativa viável ao emprego de câmeras na construção de mapas por robôs e propõem um

algoritmo para exploração de locais desconhecidos que não necessita percorrer todos os pontos do ambiente para a construção do mapa, devido ao alcance da detecção de obstáculos dos sensores empregados. Posteriormente, outro trabalho de Elfes (1987) apresenta a utilização das informações registradas na Grade de Evidência com a finalidade de permitir que robôs naveguem pelo ambiente mapeado e o conceito de níveis de processamento necessários para se obter um robô móvel autônomo. A necessidade de processos autônomos é justificada pela exposição de diversas aplicações na área de robótica, as quais exigem que os robôs evoluam seu processo de deslocamento de apenas em áreas estruturadas para locais desconhecidos e não estruturados.

A representação métrica de ambientes físicos utilizando-se Grades de Evidência – acrescidas de modificações ou não – é amplamente estudada e empregada na robótica móvel. A pesquisa de Kato, Murakami e Ohnishi (2002) evolui a estratégia de mapeamento em duas dimensões para três dimensões. Para tal, divide o ambiente a ser mapeado em um conjunto de cubos elementares, cada cubo recebendo uma classificação segundo seu estado, vazio, ocupado ou desconhecido. O trabalho apresenta também um método para o planejamento de trajetória que converte a representação métrica novamente em duas dimensões, a fim de permitir que o robô atinja um objetivo movendo-se através do espaço representado por um plano cartesiano.

O trabalho de Wolf e Sukhatme (2003) apresenta a possibilidade de registro na Grade de Ocupação da probabilidade de se encontrar um obstáculo em determinada posição, devido às imperfeições inerentes às leituras realizadas pelos sensores. O trabalho apresenta também a representação de mapas através da utilização de duas Grades de Ocupação, uma utilizada para o registro de obstáculos fixos no ambiente e outra utilizada para obstáculos dinâmicos, que são utilizadas em conjunto durante o processo de planejamento de trajetória.

A pesquisa de Gürtler, Vajta e Nagy (1997) introduz a possibilidade de representação de diferentes características dos objetos que compõem o ambiente a ser mapeado, através da

atribuição de diferentes pesos às diferentes estruturas encontradas pelos sensores do robô, como, por exemplo, paredes, caminhos livres, obstáculos e cruzamentos de vias de deslocamento. Desta forma, a estratégia de planejamento de trajetória calcula um caminho de movimentação possível com base em tais pesos registrados no mapa, após reduzir as dimensões do robô para um ponto e ampliar proporcionalmente as dimensões dos objetos presentes no ambiente. Durante o processo de navegação, a trajetória calculada como uma linha é expandida para as dimensões reais do robô e este deve utilizar seu conjunto de sensores apenas para confrontar as informações já registradas a fim de evitar colisões com obstáculos móveis.

A técnica de mapeamento geométrica também é empregada em ambientes de grande extensão, internos e externos. Marco, Prattichizzo e Vicino (2000) propõem uma solução para a redução da complexidade na representação de tais mapas através da exclusão de áreas consideradas desinteressantes da estrutura de mapeamento empregada, chamada de “Height Fields”.

O trabalho de Brezetz et al. (1996) emprega como estrutura de registro de dados o Filtro Estendido de Kalman, no qual são assinalados alguns pontos especiais que desempenham a função de marcação do ambiente. Propõe-se que o robô, durante o seu processo de deslocamento pelo ambiente, selecione alguns pontos especiais no local de modo que estes sejam utilizados como referência no cálculo do seu posicionamento relativo no espaço. Tais pontos especiais devem ser amplamente visíveis e não ambíguos. A pesquisa aborda o mapeamento de áreas externas e seleciona os maiores picos visíveis pelo robô para demarcação do ambiente. Argumenta ainda que em espaços internos a técnica pode ser empregada utilizando-se objetos de destaque presentes no local, como por exemplo, portas.

Brooks (1985) desenvolve um dos primeiros trabalhos com o objetivo de se construir mapas topológicos do ambiente. O trabalho representa o local mapeado utilizando Caminhos Livres

(“Freeways”) conectando regiões distintas do espaço e acrescenta a tal estratégia registros probabilísticos das incertezas nas informações enviadas pelos sensores e das incertezas no exato posicionamento do robô no ambiente. Desta forma, o autor propõe a adoção de Regiões Possíveis de posicionamento nas extremidades das vias livres de conexões entre espaços no ambiente.

O trabalho de Innocenti et al. (1994) evolui esta representação topológica através da descrição do ambiente como um conjunto de pontos especiais (chamados de “View Points”) que o robô pode alcançar durante suas movimentações. A posição de tais pontos especiais é obtida através de coordenadas polares relativas, referenciadas nos outros pontos especiais circundantes. A cada ponto especial também são adicionadas informações sobre seu posicionamento relativo em relação a marcas posicionadas no ambiente, de forma a não se empregar um sistema de referência absoluto.

Outra técnica largamente empregada na representação topológica de ambientes recebe o nome de Diagramas de Voronoi (“Generalized Voronoi Graph – GVG”). Tal técnica consiste em representar o ambiente utilizando-se de vias que servem de guias nas conexões entre os espaços livres do local explorado. Onde houver cruzamento de duas ou mais vias de conexão, destaca-se um ponto de cruzamento. Pode-se citar a pesquisa de Jennings, Watts e Tanis (1998) como exemplo da utilização desta técnica na construção de mapas de ambientes desconhecidos por times de robôs. Cada agente do time explora independentemente um local distinto dentro do ambiente a ser mapeado e, ao se encontrarem, podem trocar informações sobre áreas já exploradas com o objetivo de ampliarem o mapa GVG previamente obtido. A estratégia apresentada tem função de evoluir a anterior, na qual, quando um robô encontrava outro durante o processo de mapeamento do local, considerava-o como um obstáculo presente no espaço, o que introduzia deficiências na representação do ambiente.

Indica-se como referência, a pesquisa de Thrun (2002) que apresenta a relação de diversos trabalhos realizados na área de mapeamento robótico que utilizam representações métricas ou topológicas na descrição dos espaços físicos. O trabalho destaca também a adoção de ferramentas que possibilitam o registro das probabilidades relacionadas às possíveis discrepâncias do conjunto de sensores utilizado pelo robô.

Alguns trabalhos, no entanto, apresentam alternativas a estas pesquisas clássicas de mapeamento robótico, motivados, principalmente, pela diminuição da complexidade computacional envolvida no processo. Pode-se citar, por exemplo, Zimmer (2000) que utiliza uma rede neural artificial para o registro das informações coletadas pelos sensores, na qual os neurônios têm correlação com espaços no ambiente mapeado. O trabalho permite que o robô construa inicialmente um modelo primário de todo o ambiente no qual está inserido e, posteriormente, um modelo detalhado de locais críticos ou de maior importância para o seu deslocamento. Em locais onde é necessário maior detalhamento, o processo registra maior quantidade de informação em um maior volume de neurônios, em locais onde esta necessidade é menor, há um menor registro de informações e uma menor utilização de neurônios. O autor argumenta que o desenvolvimento da metodologia apresentada se deve ao fato de que, em execuções reais, o mapeamento completo e detalhado do ambiente inviabiliza a aplicação devido à sua complexidade computacional, porém existem locais nos quais o robô necessita de grande informação sobre o ambiente. Assim, o ambiente em geral é mapeado com menor número de detalhamento e, nos locais de manobra, por exemplo, o ambiente é modelado com grande quantidade de informação.

Também motivado pelas dificuldades computacionais que implementações reais têm em lidar com o planejamento de trajetória utilizando mapas geométricos, o trabalho de Chow, Xu e Tso (2002) mapeia diretamente os comandos enviados para o sistema de deslocamento do

veículo de pesquisa conforme as informações coletadas pelo conjunto sensor. Esse mapeamento é realizado por uma rede neural artificial, porém possui a restrição de não apresentar um desempenho tão eficiente em espaços abertos, devido à menor quantidade de informação sobre obstáculos recebidas pelos sensores.

### **2.3 Exploração de ambientes desconhecidos**

Os primeiros trabalhos realizados na área de pesquisa de exploração de ambientes desconhecidos não tinham como objetivo possibilitar que o robô desenvolvesse uma trajetória de deslocamento estratégico pelo espaço ainda não mapeado a fim de poder registrar informações deste ambiente. O objetivo principal era somente permitir que o agente robótico fosse capaz de se deslocar por um ambiente desconhecido sem colidir com os obstáculos presentes. Rao et al. (1987) aponta que os primeiros processos desenvolvidos para a aquisição de terreno eram realizados de maneira acidental durante a navegação do robô por áreas já conhecidas e Yamauchi (1997) relaciona diversas deficiências destas técnicas iniciais de exploração de ambientes, como: (i) o emprego de técnicas puramente reativas para controlar o deslocamento do robô. Tais técnicas possibilitam que as detecções realizadas pelo conjunto sensor sejam diretamente convertidas em ações como, por exemplo, a execução de trajetórias que apenas permitam movimentações acompanhando uma parede detectada pelos sensores do agente robótico; (ii) a aplicação em ambientes cujos obstáculos existentes estejam dispostos somente em uma configuração paralela ou perpendicular ao deslocamento do robô; (iii) e a restrição de que os ambientes a serem explorados fossem locais simples, essencialmente não obstruídos, e que os obstáculos existentes não estivessem dispostos de forma complexa.

Diversos trabalhos realizados necessitam de algum conhecimento prévio do local a ser mapeado antes que o robô inicie o processo de exploração.

O trabalho de Lumelsky, Mukhopadhyay e Sun (1990) apresenta duas estratégias bem definidas para a exploração de ambientes desconhecidos que empregam uma técnica de contorno dos obstáculos existentes. A primeira técnica apresenta o requisito de que todos os obstáculos do espaço a ser explorado sejam mutuamente visíveis pelo conjunto de sensores do robô utilizado. A segunda técnica proposta assume que o ambiente tenha formato retangular, de dimensões conhecidas e que seja possível explorá-lo inteiramente segundo faixas paralelas de dimensões aproximadamente iguais.

Zelinsky (1992) apresenta uma pesquisa na qual o robô pode-se deslocar desde uma posição inicial até uma posição final, chamada de objetivo, sem que o espaço entre estas duas posições seja conhecido. A estratégia assume que a trajetória entre a posição corrente do robô e objetivo esteja livre de obstáculos e, caso algum obstáculo seja encontrado, o mapa é atualizado. Assim, diversos objetivos podem ser atribuídos ao robô de forma que este construa um completo mapa do local. O algoritmo empregado divide o ambiente em células e atribui pesos para cada uma delas conforme sua distância da posição final e, para tanto, necessita do conhecimento prévio das dimensões do local a ser explorado e das posições relativas dos objetivos a serem alcançados e da posição do início de trajetória do robô.

O trabalho de Burlina, DeMenthon e Davis (1992) também possibilita que o robô se desloque por um ambiente desconhecido através do conhecimento de sua posição atual e da posição de objetivos. A definição do percurso de movimentação considera informações probabilísticas de colisão com obstáculo e alcance do objetivo e pode ser alterada de acordo com o nível de risco aceitável para dado deslocamento do agente robótico.

Como alternativa aos trabalhos clássicos de exploração de ambientes, pode-se citar a pesquisa de Batalin e Sukhatme (2003), que propõe que o processo seja auxiliado por transmissores de

Rádio Freqüência (RF). O robô, durante o seu deslocamento pelo ambiente, deposita no terreno os transmissores de RF em lugares previamente definidos pelo algoritmo proposto. Tais transmissores operam como marcos no ambiente explorado e permitem que o robô conheça sua posição relativa, sem que seja necessária a execução de outros procedimentos de localização.

A pesquisa de Acar e Choset (2001) possibilita ao robô explorar inteiramente o ambiente no qual está inserido sem o conhecimento prévio de informações sobre o local, através de uma trajetória padrão de ziguezague. A trajetória padrão pode ser alterada de acordo com a detecção de configurações específicas dos obstáculos presentes no ambiente os quais, por exemplo, impossibilitem a continuidade do percurso inicialmente estabelecido. O trabalho argumenta ainda que os algoritmos desenvolvidos para este tipo de exploração são complicados e exigem grande capacidade de processamento, principalmente se o robô que for explorar o ambiente não possuir nenhuma informação sobre as dimensões deste ambiente.

#### **2.4 Cobertura completa de ambientes por robôs móveis**

Cobertura completa de ambientes é um caso particular do processo de planejamento de trajetória no qual o robô deve percorrer todos os espaços livres e alcançáveis no ambiente no qual está inserido (em inglês este processo é chamado de “entire region filling” ou “environment complete coverage”) e, anteriormente à sua aplicação em processos de mapeamento robótico, seu emprego ocorria principalmente em agentes robóticos destinados a funções específicas, como detecção de minas terrestres, tarefas de limpeza e aparagem de grama. Estas aplicações iniciais geralmente utilizavam um mapa previamente construído do ambiente de trabalho do robô registrado na memória de tal robô.

O trabalho de Yasutomi (1988) apresenta uma estratégia de cobertura total de um determinado espaço por robôs de limpeza. A aplicação da estratégia apresentada ocorre apenas em locais de pouca complexidade, como um ambiente vazio com apenas um obstáculo no centro. O robô emprega um mapa previamente construído do local de trabalho no qual o espaço é dividido em células livres e ocupadas e, durante a execução da tarefa de limpeza, estas células são classificadas como já visitadas e não visitadas. No processo de navegação, o robô também utiliza sonares para evitar colisões com obstáculos presentes no ambiente.

A pesquisa de Carvalho et al. (1997) apresenta um método que permite a um robô móvel utilizar o mapa do local previamente construído com o objetivo de percorrer todo o ambiente adotando uma trajetória em ziguezague. A estratégia apresenta um comportamento de mudança de tamanho das faixas paralelas que descrevem o percurso do robô quando algum obstáculo presente no ambiente o impede de realizar uma trajetória em ziguezague perfeita, chamado de “shift size”. A abordagem apresenta restrições de eficiência de movimentação em ambientes com configurações de mínimos locais – chamados de “beco sem saída” – e em situações onde a disposição dos obstáculos não é coerente com o mapa construído.

Schmidt e Hofner (1998) apresentam um trabalho desenvolvido para robôs de limpeza que também emprega um mapa previamente construído do ambiente de trabalho para possibilitar ao robô percorrer todas as áreas livres deste local. O algoritmo permite ao robô explorar locais mais complexos, calcular trajetórias alternativas durante o deslocamento que tenham por objetivo evitar colisões com obstáculos móveis não mapeados e informar a um operador as áreas que eventualmente não foram cobertas ao término de sua operação.

A pesquisa de cobertura completa de região realizada por Luo, Yang e Meng (2002) permite ao robô percorrer todos os pontos do espaço em que está inserido em uma trajetória em ziguezague, necessitando apenas que este robô tenha conhecimento prévio das dimensões da área a ser preenchida. A técnica empregada utiliza uma rede neural na qual cada neurônio é

relacionado a um ponto do espaço a ser preenchido e recebe um peso diferente. Estes pesos são determinados com bases nas áreas a eles relacionadas. Áreas já limpas e áreas que contêm obstáculos recebem pesos que repelem o robô, áreas ainda não limpas recebem pesos que atraem o robô.

Com a finalidade de mapear ambientes desconhecidos, o trabalho de González et al. (1996) propõe dois tipos diferentes de planejamento de trajetória: a primeira com o objetivo de aquisição das configurações limites de uma porção do terreno no qual o robô está inserido e a segunda com o objetivo de percorrer inteiramente a região delimitada a fim de mapear o ambiente. O primeiro tipo de planejamento de trajetória constrói as chamadas Regiões Complementares, ou seja, trechos elementares e desconhecidos do espaço conectados a regiões conhecidas e já mapeadas. O novo ambiente detectado é completamente preenchido pelo robô empregando-se o segundo tipo de planejamento de trajetória, a qual adota percursos em ziguezague. Durante este percurso as diferentes porções de espaço são classificadas em livres, ocupadas e parcialmente ocupadas no mapa construído.

Também com o objetivo de mapear ambientes, a pesquisa de Fu e Lang (1999) emprega lógica nebulosa para controlar o deslocamento do robô. A lógica nebulosa tem como parâmetros de entrada as distâncias de obstáculos à frente, à esquerda e à direita e seu conjunto de regras são fundamentadas em experiências reais de controle no deslocamento de robôs. O trabalho utiliza ainda uma máquina de estados finita para ativar os diferentes padrões de comportamento que o controle nebuloso deve realizar, de forma que o comportamento do robô realize a exploração uma região desconhecida até percorrê-la inteiramente, voltando em seguida em busca de locais ainda não explorados. Entretanto, a proposta inicial está restrita apenas a ambientes simples.

### 3. O MODELO PROPOSTO

O modelo proposto no presente trabalho para mapeamento de ambientes desconhecidos e controle do deslocamento de exploração do robô nestes ambientes é apresentado na figura 1.

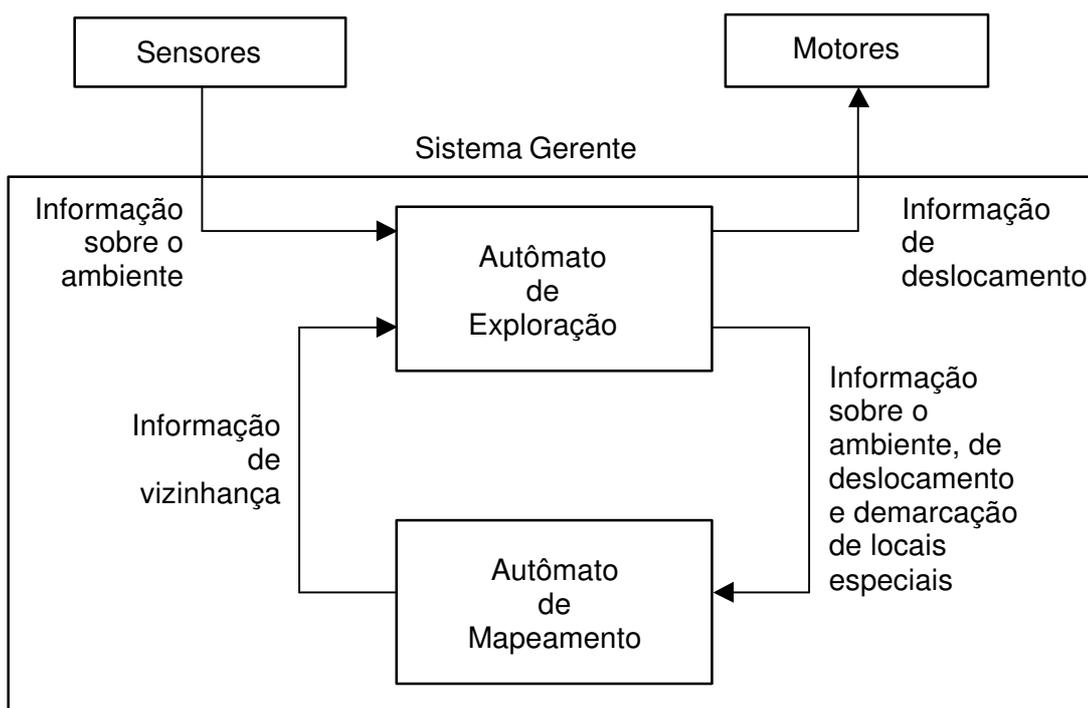


Figura 1. O modelo de fluxo de dados proposto pelo presente trabalho.

Como escopo deste trabalho foca o sistema de mapeamento e exploração de ambientes, assume-se que as informações provenientes dos sensores do robô são precisas e livres de ruído e os deslocamentos efetuados pelo seu sistema motor são acurados.

O Autômato de Exploração tem como função determinar a direção do próximo deslocamento efetuado pelo sistema motor do robô durante a exploração de um ambiente desconhecido. Esta decisão é feita com base nos dados coletados pelo sistema sensor sobre a existência ou não de obstáculos no perímetro circundante ao robô, nas informações previamente registradas pelo Autômato de Mapeamento e no algoritmo de exploração executado por este autômato.

O Sistema Gerente coleta as informações sobre o ambiente capturadas pelo conjunto de sensores e as ordena de modo a formar cadeia de entrada do Autômato de Exploração. Este Sistema Gerente também é responsável por coletar os dados já registrados sobre a topologia da vizinhança previamente mapeada pelo robô – se já houver sido registrada alguma informação – e os ordenar, em seqüência posterior aos dados capturados pelos sensores, na cadeia de entrada do Autômato de Exploração. A informação de saída indicada pela execução deste autômato determina qual a direção do próximo movimento a ser executado pelo sistema motor do robô e tal informação é transmitida para o Autômato de Mapeamento, além de também ser transmitida para o sistema motor.

O Sistema Gerente também tem como função construir a cadeia de entrada do Autômato de Mapeamento com as informações coletadas pelo sistema sensor, a informação de deslocamento e a informação de demarcação de locais especiais do ambiente, indicadas pelo Autômato de Exploração. O Autômato de Mapeamento é responsável pelo registro dos dados sobre a topologia do ambiente que está sendo mapeado pelo robô.

A demarcação de locais especiais é um processo realizado pelo Autômato de Exploração, no qual são determinadas algumas posições relevantes no ambiente que está sendo explorado. Tais posições são calculadas de acordo com a configuração dos obstáculos no espaço que está sendo explorado e são chamadas de marcos. Esta demarcação do ambiente é realizada com dois objetivos: o primeiro é facilitar o processo de construção de uma trajetória de retorno de um local já conhecido que envolva um menor número de deslocamentos do robô durante o processo de exploração; e o segundo é auxiliar na futura utilização do mapa construído durante o processo de navegação do robô e, para tanto, o Autômato de Mapeamento também é responsável pelo registro de tais locais especiais em sua representação do ambiente mapeado.

## **4. AUTÔMATOS ADAPTATIVOS**

Este capítulo apresenta a ferramenta adaptativa utilizada para os processos de mapeamento e exploração de ambientes, os Autômatos Adaptativos.

O formalismo dos Autômatos Adaptativos é aplicado em diversas áreas de pesquisa científica e tecnológica. Pode-se destacar as seguintes áreas de pesquisa: (i) linguagens formais e linguagens de computação – na qual tem-se, por exemplo, o trabalho de Freitas e Neto (2001) que apresenta uma ferramenta para o auxílio na implementação de aplicações que utilizam multilinguagens, isto é, aplicações que empregam mais de uma linguagem de programação para a execução de uma determinada tarefa; (ii) processamento de linguagem natural – na qual pode-se citar como exemplo a pesquisa de Menezes e Neto (2002), que propõe um autômato adaptativo para ser treinado com uma base de dados com a finalidade de classificar informações lingüísticas de um dado corpus e a pesquisa de Neto e Moraes (2002), que propõe um modelo para o tratamento das ambigüidades inerentes à linguagem natural com base na adaptabilidade deste formalismo; e (iii) sistemas de decisão e aprendizagem de máquina – na qual pode-se citar o trabalho de Pistori e Neto (2002) que apresenta o AdapTree, um sistema que possibilita a construção de árvores de indução.

Todos estes exemplos utilizam, essencialmente, a mesma ferramenta adaptativa que é empregada em nossa abordagem de mapeamento e exploração de ambientes.

### **4.1 Introdução**

Para apresentar a teoria dos Autômatos Adaptativos (seção 4.4), este capítulo aborda brevemente os fundamentos que introduzem este formalismo: autômatos finitos (seção 4.2) e

autômatos de pilha (seção 4.3), de forma a apresentar os dispositivos que reconhecem linguagens do tipo 1, 2 e 3 na hierarquia de Chomsky<sup>1</sup>, como ilustra a figura 2 (NETO, 1987).

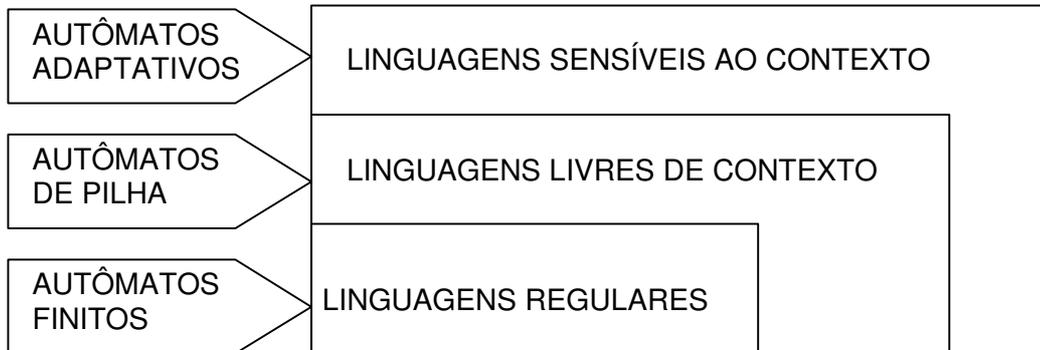


Figura 2. Hierarquia de Chomsky para linguagens tipo 1, 2 e 3.

## 4.2 Autômatos Finitos

Essencialmente, definem-se autômatos finitos como dispositivos reconhecedores de linguagens, que conseguem responder a perguntas do tipo: “dada uma cadeia  $w$ ,  $w$  é membro da linguagem  $L$ ?”, sendo, assim, especialmente projetados para a linguagem  $L$  em questão. Autômatos finitos possuem estados internos distintos e entrada composta pelos símbolos da cadeia  $w$  a ser reconhecida. A leitura de cada símbolo da cadeia  $w$  possibilita a mudança do estado interno do autômato, de acordo com o estado interno corrente, símbolo de entrada e a função de transição de estados do autômato.

### 4.2.1 Autômatos finitos determinísticos

Autômatos finitos determinísticos formam uma subclasse dos autômatos finitos, portanto seguem os fundamentos descritos anteriormente e possuem, como característica, sua operação inteiramente determinada pela entrada lida e pela sua configuração corrente.

<sup>1</sup> A hierarquia de Chomsky prevê ainda uma linguagem do tipo 0 que engloba os outros três tipos de linguagens citadas, mas que não faz parte do escopo deste trabalho.

A mudança de estados em um autômato finito determinístico é estabelecida por uma função de transição própria do autômato, pelo estado atual de operação e pelo símbolo corrente em sua cadeia de entrada, os quais determinam um único estado para o qual a mudança ocorrerá.

O autômato pode operar quantas vezes for necessário para o consumo da cadeia de entrada. Essa seqüência é chamada de uma computação executada pelo autômato sobre a cadeia de entrada, e tem comprimento  $n$ , que representa o número de passos necessários para o consumo de tal cadeia. A cadeia de entrada é aceita pelo autômato finito determinístico quando após todos os movimentos internos executados, o último símbolo da cadeia seja consumido e o último estado do autômato seja um estado final.

O Anexo A contém a definição formal dos autômatos finitos determinísticos.

#### **4.2.2 Autômatos finitos não-determinísticos**

Os autômatos finitos não-determinísticos compõem a outra subclasse dos autômatos finitos, portanto seguem os fundamentos descritos na seção introdutória e possuem como característica diferenciadora o fato de que a mudança de estado é só parcialmente determinada pelo estado corrente do autômato e pelo símbolo de entrada. Há, portanto, um grupo de estados possíveis para uma dada transição e sua escolha não é determinada por nenhum modelo.

Autômatos finitos não-determinísticos podem também exibir transições em vazio, isto é, o autômato pode se mover de um estado para outro sem consumir nenhum símbolo da cadeia de entrada. Esta transição em vazio é representada pela letra  $\epsilon$ . Por fim, pode-se ter, nesta subclasse dos autômatos finitos, o consumo de mais de um símbolo da cadeia de entrada antes da mudança de estado.

O Anexo B contém a definição formal dos autômatos finitos não-determinísticos.

### **4.2.3 Propriedades das linguagens aceitas por autômatos finitos**

As linguagens reconhecidas por autômatos finitos são chamadas de linguagens regulares, isto é, a menor classe de linguagem existente que inclui apenas o símbolo vazio  $\epsilon$ , a concatenação dos símbolos pertencentes à linguagem e a concatenação de sentenças, também regulares, formada por tais símbolos (LEWIS; PAPADIMITRIOU, 1981).

## **4.3 Autômatos de Pilha e Linguagens Livres de Contexto**

Comparando-se com os autômatos finitos apresentados anteriormente, autômatos de pilha compõem uma classe de autômatos mais abrangente, os quais, portanto, podem reconhecer uma classe de linguagem mais abrangente, chamadas linguagens livres de contexto.

### **4.3.1 Linguagens livres de contexto**

Adicionando-se às linguagens regulares a possibilidade de aninhamento sintático, isto é, o agrupamento de expressões tão complexas quanto às expressões externas ao agrupamento, obtém-se outra classe de linguagem mais abrangente, as linguagens livres de contexto (LEWIS; PAPADIMITRIOU, 1981).

### **4.3.2 Autômatos de pilha**

Para que um autômato reconheça uma linguagem livre de contexto deve-se adicionar à sua estrutura uma memória de registro. Esta memória de registro é organizada em pilha, isto é, o

último símbolo registrado é o primeiro símbolo a ser recuperado, pois os aninhamentos sintáticos que estão mais ao centro da expressão fecham-se primeiro do que os que estão na periferia da expressão.

O Anexo C contém a definição formal dos autômatos de pilha.

#### **4.4 Autômatos Adaptativos e Linguagens Sensíveis ao Contexto**

A classe mais abrangente de autômatos que este trabalho trata é a classe dos Autômatos Adaptativos. Esta classe corresponde à classe dos reconhecedores de linguagens sensíveis ao contexto, além de também reconhecer as linguagens regulares e livres de contexto já expostas.

##### **4.4.1 Linguagens sensíveis ao contexto**

Linguagens sensíveis ao contexto possuem, em sua formação, regras de substituição condicionais, nas quais a existência de um símbolo está restrita a situações de dependência do contexto. Desta forma, pode-se ter, por exemplo, diversas interpretações possíveis de um mesmo caractere dependendo do ponto onde ele está inserido ou de outros caracteres que estão ao redor.

##### **4.4.2 Autômatos Adaptativos**

Autômatos Adaptativos foram propostos por Neto (1993) como uma forma eficiente de se executar o reconhecimento da classe de linguagens sensíveis ao contexto. A computação de tais linguagens se dá pela existência de funções adaptativas presentes no autômato, as quais podem modificar sua configuração de acordo com o processamento da cadeia de entrada,

inserindo e eliminando transições adequadamente, de forma que este possa adquirir nova estrutura.

No início do processamento tem-se uma versão inicial automodificadora do autômato que, conforme processa a cadeia de entrada, exibe configurações intermediárias, diferentes da sua configuração original, até que uma configuração final é alcançada, quando a cadeia de entrada é totalmente consumida.

A implementação de uma função adaptativa é chamada de ação adaptativa, e compreende um grupo de ações adaptativas elementares, que podem ser classificadas em três tipos:

- Consulta: busca na configuração corrente do autômato por um padrão de produção específico, retornando valores indefinidos na especificação.
- Eliminação: modifica a configuração do autômato através da exclusão de uma determinada produção do conjunto corrente.
- Inclusão: modifica a configuração do autômato através da inserção de uma determinada produção no conjunto corrente.

Ações adaptativas podem ser classificadas em dois tipos:

1. Anterior: a ação adaptativa é efetuada sempre imediatamente antes de uma produção ocorrer.
2. Posterior: a ação adaptativa é efetuada sempre imediatamente após uma produção ocorrer.

Cada ação adaptativa pode conter, opcionalmente:

1. Chamada de uma função adaptativa inicial: a ação adaptativa correspondente deve ser executada anteriormente a qualquer consulta ou alteração indicada pelo conjunto de ações adaptativas elementares.

2. Chamada de uma função adaptativa final: a ação adaptativa correspondente deve ser executada imediatamente após todas as consultas ou alterações indicadas pelo conjunto de ações adaptativas elementares.

Em uma função adaptativa pode-se ter, também:

1. Geradores: identificadores que recebem um valor único, isto é, nunca antes utilizado, no início da ação adaptativa e permanecem com este valor até o final da ação.
2. Variáveis: identificadores que recebem um valor de acordo com o resultado de ações adaptativas elementares de consulta ou exclusão.

Formalmente, definem-se Autômatos Adaptativos por:  $M = \{Q, \Sigma, P, q_0, F, \Phi\}$

com:

Q: conjunto de estados do autômato;

$\Sigma$ : alfabeto de entrada;

P: função de transição de estados  $P = Q \times \Sigma \rightarrow Q \times \Phi^*$ ;

$q_0$ : estado inicial único  $q_0 \in Q$ ;

F: conjunto de estados finais  $F \subseteq Q$ .

$\Phi$ : conjunto de funções adaptativas que podem ser definidas como uma ênupla do tipo:

$(F, P, V, G, C, E, I, A, B)$

com:

F: nome da função adaptativa;

P: lista de parâmetros formais ( $r_1, r_2, \dots$ );

V: conjunto de identificadores de variáveis;

G: conjunto de identificadores de geradores;

C: conjunto de padrões das produções a serem consultadas;

E: conjunto de padrões das produções a serem localizadas e eliminadas;

I: conjunto de padrões das produções a serem inseridas;

A: ação adaptativa inicial;

B: ação adaptativa final.

Uma ação adaptativa corresponde à chamada de uma função adaptativa, e é dada pelo par ordenado (F, P):

F: nome da função adaptativa;

P: seqüência de argumentos ( $p_1, p_2, \dots$ ) que serão utilizados pela função adaptativa F, e designam valores a serem utilizados em substituição aos correspondentes parâmetros formais para a composição de transições adaptativas a serem inseridas, consultadas ou eliminadas do autômato.

A declaração da função de F com n parâmetros q consiste em um cabeçalho:

$$F (q_1, q_2, \dots, q_n)$$

e um corpo, com o seguinte formato:

$$\{ \quad \text{declaração de nomes (opcional)} \\ \quad \text{declaração de ações (opcional)} \quad \}$$

No qual a declaração de nomes é uma lista de nomes escolhidos para identificar as variáveis e geradores do escopo da função, e declaração de ações é uma lista de ações adaptativas elementares, opcionalmente precedida e seguida pelas chamadas de funções adaptativas.

A forma de uma declaração de nomes assume o seguinte aspecto:

$$v_1, v_2, \dots, v_m, g_1^*, g_2^*, \dots, g_n^*$$

Na qual os nomes seguidos por asteriscos denotam geradores e os nomes restantes denotam variáveis.

Parâmetros também recebem valores antes da execução da função. Cada parâmetro assume o valor corrente ao argumento de posição correspondente da chamada de função, permanecendo inalterado durante toda a execução desta função.

Declaração de ações é uma lista de ações adaptativas elementares, as quais indicam as consultas e alterações a serem realizadas sobre a configuração corrente do autômato adaptativo. Estas ações adaptativas elementares possuem três tipos distintos:

1. Ações adaptativas elementares de consulta:                   ? [produção];
2. Ações adaptativas elementares de eliminação:               - [produção];
3. Ações adaptativas elementares de inclusão:                   + [produção];

As transições de estado têm sua representação textual denotadas por:

$$(g, e, s): A, \rightarrow (g', e', s'), B$$

com:

e: estado-origem da transição;

e': estado-destino da transição;

g (opcional): conteúdo da pilha antes da transição;

g' (opcional): conteúdo da pilha depois da transição;

s (opcional): símbolo consumido pela transição descrita pela produção;

s' (opcional): símbolo incluído na cadeia de entrada como resultado da transição descrita pela produção;

A (opcional): ação adaptativa anterior;

B (opcional): ação adaptativa posterior.

E, no caso particular, onde  $g = g' = s' = \varepsilon$ , tem-se:

$(e, s): A, \rightarrow e', B$

A figura 3 ilustra a representação gráfica da transição:

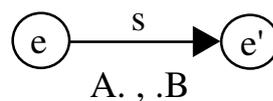


Figura 3. Representação gráfica da transição dos Autômatos Adaptativos.

O trabalho de Neto (1993) é indicado como uma referência completa e detalhada sobre o formalismo apresentado.

## 5. AUTÔMATO DE MAPEAMENTO

Este capítulo apresenta a estrutura empregada para o registro do mapa do ambiente explorado e, para tanto, apresenta no item 5.1 as características gerais do modelo proposto comparadas com as características dos outros modelos de representação de ambientes. Apresenta, também, a configuração inicial do Autômato de Mapeamento no item 5.2 e sua operação durante o deslocamento do robô pelo ambiente no item 5.3. O item 5.4 detalha a configuração do Autômato de Mapeamento e, por fim, algumas considerações sobre o modelo proposto são feitas no item 5.5. Este Autômato de Mapeamento foi apresentado no trabalho de Sousa, Hirakawa e Neto (2004b) e o Apêndice A contém sua descrição algébrica completa, de acordo com a notação apresentada no capítulo 4, sugerida na pesquisa de Neto (1993), exceto pela representação do símbolo vazio  $\epsilon$  com o caractere #.

### 5.1 Introdução

O modelo proposto para construção de mapas de ambientes consiste em um Autômato Adaptativo que registra a topologia do ambiente através de alterações em sua estrutura. Este autômato apresenta uma configuração inicial simples e pequena, a qual é expandida durante a exploração do local a ser mapeado pelo robô. Tal configuração inicial é modelada de forma que sua estrutura adquira a representação do ambiente segundo informações coletadas pelo conjunto de sensores. Assim, o processo de mapeamento se dá de acordo com a seguinte seqüência de eventos: (i) o ambiente é sondado, (ii) as informações coletadas pelos sensores são registradas no Autômato de Mapeamento e (iii) o robô desloca-se para a próxima posição para repetir o processo. A seqüência adotada é comum a outras estratégias de reconhecimento desenvolvidas, como por exemplo, no trabalho apresentado por Howell e Donald (2000) que

utiliza processo semelhante de sondagem, registro e deslocamento, e a pesquisa de Indelicato et al. (1997) que utiliza um método similar no tratamento de informações dos sensores ultra-sônicos para o reconhecimento geométrico de obstáculos e referências no ambiente. A diferença entre os exemplos citados e a proposta apresentada neste trabalho é verificada na escolha da próxima posição para a repetição do processo, para qual o robô deve se deslocar. Enquanto o primeiro elege posições distantes como objetivo de seu deslocamento (a eleição de posições distantes está relacionada com o emprego de sensores ultra-sônicos que equipam o robô específico do trabalho citado), o segundo seleciona obstáculos identificados como referência para o seu processo de movimentação. A abordagem apresentada no presente trabalho, devido ao fato de não estar restrita a nenhum tipo de sensor específico, adota que o conjunto sensor fornece dados sobre a existência ou não de obstáculos no ambiente ao redor do robô dentro de uma distância igual às dimensões do próprio robô. Este princípio impõe que a distância de deslocamento do robô durante o movimento de exploração seja igual à dimensão deste robô, ou seja, em cada passo de movimentação o robô se desloca uma distância equivalente à sua própria extensão.

Semelhante a outros trabalhos realizados na área de robótica móvel, adotou-se, no presente trabalho, que o conjunto sensor fornecerá ao robô informações sobre a configuração do ambiente ao seu redor nas quatro direções cardinais – norte, sul, leste e oeste. Cita-se, por exemplo, a pesquisa de Lumelsky, Mukhopadhyay e Sun (1990), que define as mesmas quatro direções com base de alinhamento para os eixos de referência no seu processo de construção de mapas; o trabalho de Batalin e Sukhatme (2003), que também emprega as quatro direções cardinais para marcação da condição do local relacionado a um instante de tempo em seu processo de exploração de ambientes; e o trabalho de Zelinsky (1992), que utiliza sensores para coleta de dados sobre a existência ou não de obstáculos no ambiente nas quatro direções ortogonais ao movimento do robô. Este último artigo citado não especifica a nomenclatura

Norte, Sul, Leste e Oeste, e tal terminologia é utilizada no presente trabalho apenas como facilitadora na identificação das quatro diferentes direções e não implica, necessariamente, no conhecimento de informações cardinais absolutas pelo robô.

Em conjunto com a informação de direção, o Sistema Gerente coleta dos sensores a informação sobre a condição do ambiente – ocupado ou livre para deslocamento. Esta classificação também é empregada em diversas estratégias, como, por exemplo, no trabalho de Movarec e Elfes (1985), que propõe a técnica de Grades de Ocupação, na qual o ambiente é dividido em células e cada célula é preenchida com informação sobre a existência ou não de obstáculos; na pesquisa de Yamauchi (1997), a qual apresenta uma estratégia de exploração de ambientes desconhecidos e classifica o mapa construído em áreas livres e não livres; e no trabalho de González et al. (1996), que divide o ambiente de trabalho em quadrados de dimensões iguais às do robô e os classifica em livres ou ocupados. Destaca-se ainda que este último trabalho apresenta uma terceira classificação para os quadrados divisórios do ambiente, o rótulo de local parcialmente ocupado.

Ambas as informações – direção e condição – são registradas pelo Autômato de Mapeamento, que possui, ainda, uma terceira classificação de condição possível: local desconhecido, a qual é utilizada para designar locais ainda não explorados pelo agente robótico. Esta terceira classificação é comum a outros trabalhos da área de robótica móvel, como, por exemplo, o trabalho de Kato, Murakami e Ohnishi (2002), que constrói uma representação em três dimensões de uma pequena porção do espaço ao redor do robô através de sua divisão deste espaço em cubos elementares com classificação de livre, ocupado ou desconhecido e a pesquisa de Batalin e Sukhatme (2003), a qual classifica o ambiente que está sendo explorado pelo robô em aberto ou explorado.

## 5.2 Configuração Inicial do Autômato de Mapeamento

A figura 4 contém a representação gráfica da configuração inicial do Autômato de Mapeamento, segundo o padrão apresentado no capítulo 4. Tal tipo de representação foi escolhido para permitir uma clara visualização do autômato.

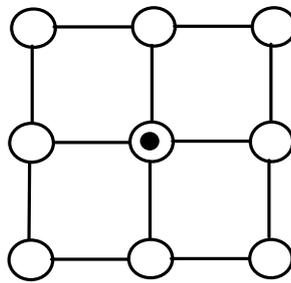


Figura 4. Representação gráfica da configuração inicial do Autômato de Mapeamento.

A configuração inicial do Autômato de Mapeamento consiste em nove estados interligados por transições específicas em forma de treliça, a qual possui três estados de lado. Tais transições específicas compõem-se de uma coleção de transições que denotam áreas ainda não mapeadas. O estado central corresponde ao estado inicial do autômato e ao ponto inicial de exploração no ambiente real a ser mapeado. O estado corrente é assinalado pelo círculo escuro (na figura 4 o estado corrente corresponde ao estado inicial do autômato).

A figura 5 apresenta a completa configuração inicial do Autômato de Mapeamento, na qual estão representadas quatro transições especiais que possuem a função de prover suporte a expansões da estrutura e transições reflexivas marcando as arestas desta estrutura.

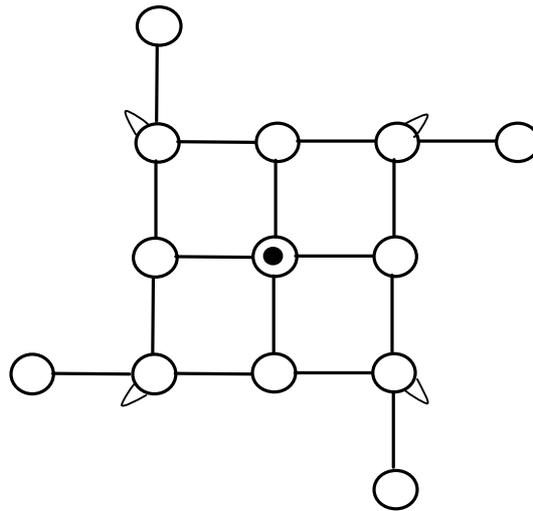


Figura 5. Configuração inicial completa do Autômato de Mapeamento.

O Sistema Gerente fornece ao Autômato de Mapeamento as informações coletadas pelo conjunto de sensores durante o deslocamento do robô pelo ambiente. Tais informações contêm indicações de direção – norte, sul, leste ou oeste – e condição – ocupado ou livre. Ao computar estes dados em sua cadeia de entrada, o autômato substitui as quatro transições adjacentes ao estado corrente que denotam áreas ainda não mapeadas por transições que representem estas informações coletadas pelo conjunto de sensores. A figura 6 apresenta todas as dezesseis diferentes possibilidades existentes para o conjunto de dados dos sensores, na qual setas duplas representam transições indicativas de áreas não obstruídas e linhas espessas representam transições indicativas de áreas obstruídas.

### 5.3 Operação do Autômato

Além das informações coletadas pelos sensores, a cadeia de entrada do Autômato de Mapeamento contém a informação de movimento definida pelo Autômato de Exploração – norte, sul, leste ou oeste – que está condicionada à existência de uma transição indicativa de área livre para deslocamento. O processamento desta informação produz uma mudança de

estado consistente no autômato e, acompanhando tal mudança, uma ampliação da estrutura do autômato na direção do movimento. A ampliação da estrutura é realizada pela execução de ações adaptativas descritas no capítulo 4, que adicionam uma linha ou uma coluna à configuração corrente, de acordo com a direção do movimento: movimentos em direção ao norte adicionam uma linha à parte superior do autômato, movimentos em direção ao sul adicionam uma linha à parte inferior do autômato, movimentos em direção ao leste adicionam uma linha ao lado direito do autômato e movimentos em direção ao oeste adicionam uma linha ao lado esquerdo do autômato. Destaca-se que “parte superior”, “parte inferior”, “lado esquerdo” e “lado direito” são assim chamados apenas por concordância com a representação gráfica retangular do Autômato de Mapeamento.

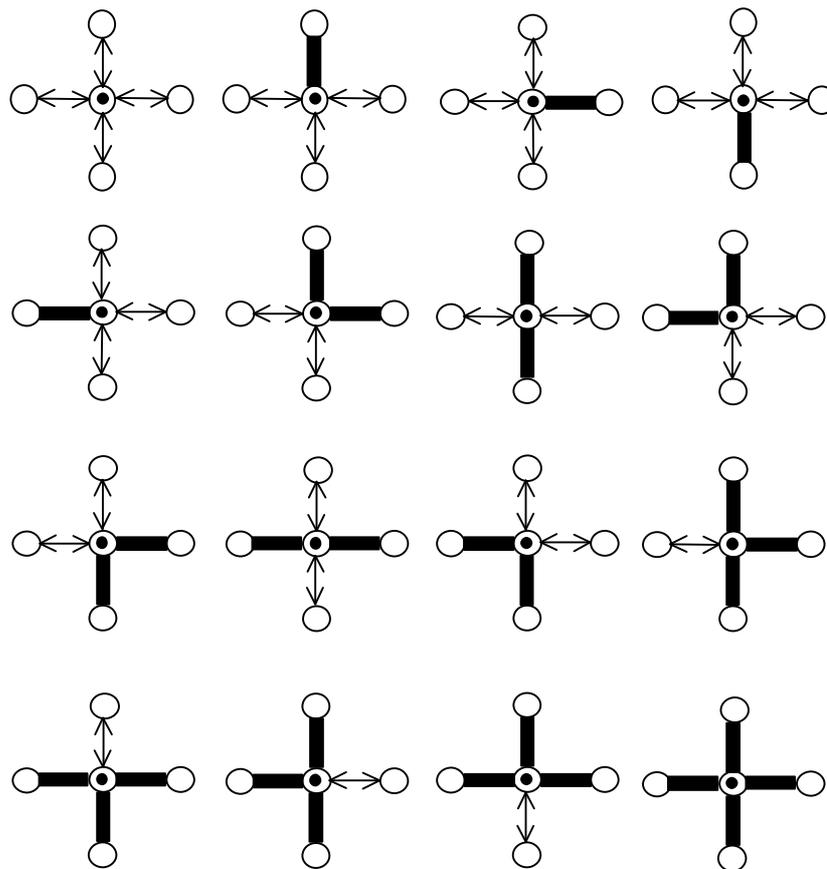


Figura 6. As dezesseis diferentes possibilidades dos dados coletados pelos sensores.

A figura 7 exemplifica o resultado de um movimento em direção ao leste após a substituição das transições que representam áreas ainda não mapeadas por transições que indicam três direções ocupadas e uma direção (leste) livre para deslocamentos. Nesta figura o processo de expansão ainda não está concluído.

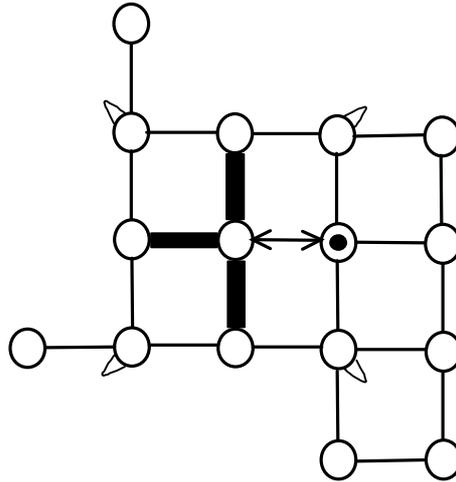


Figura 7. Expansão da configuração do autômato após um movimento em direção ao Leste.

Para concluir a operação de ampliação do autômato, as ações adaptativas reposicionam as transições reflexivas indicativas dos estados nas arestas da estrutura e criam novas transições de suporte às ampliações futuras. A figura 8 representa a conclusão de operação.

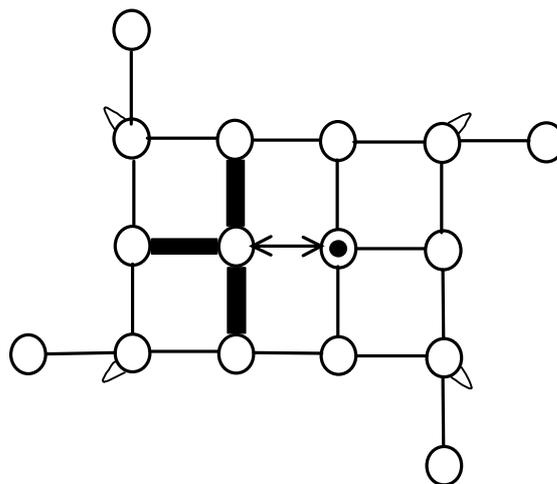


Figura 8. Conclusão do processo de ampliação da estrutura do Autômato de Mapeamento.

As ampliações na estrutura do autômato apenas ocorrem quando o estado corrente se encontra na borda da estrutura. Movimentos de retorno dentro da estrutura ou deslocamentos em direções que não levem o estado corrente para a borda do autômato não operam tais processos de expansão. Nota-se que a nomenclatura “borda” também foi adotada segundo a representação gráfica da estrutura do Autômato de Mapeamento.

O estado inicial do Autômato de Mapeamento é adotado como origem na representação do mapa. Assim, qualquer ponto do mapa é relacionado com qualquer ponto no ambiente físico pela associação de cada transição do autômato com seu deslocamento correspondente executado pelo robô no mundo real.

Por fim, a figura 9 apresenta um mapa completo com a representação das informações adquiridas pelo robô após a exploração de uma sala simples e vazia em forma de “L”. O círculo preto indica que o robô finalizou a exploração no canto superior direito do ambiente.

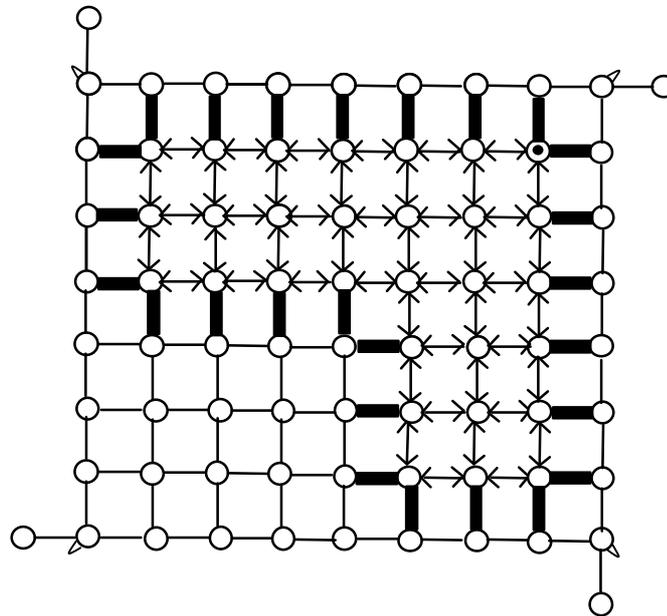


Figura 9. Exemplo de uma sala vazia inteiramente registrada no Autômato de Mapeamento.

Os dados extraídos do Autômato de Mapeamento pelo Sistema Gerente para compor parte da informação de entrada do Autômato de Exploração são apresentados na figura 10. Estes dados contêm informação sobre a condição de dois estados adjacentes ao estado corrente do

autômato nas quatro direções e é chamada de informação de vizinhança. Pode-se ter, como condições possíveis desta informação de vizinhança, caminho obstruído, caminho livre, local não mapeado ou transição ainda não criada. Esta última informação significa que o robô está próximo a uma borda do mapa pois a não existência da transição denota que o mapa ainda não foi expandido em dada direção.

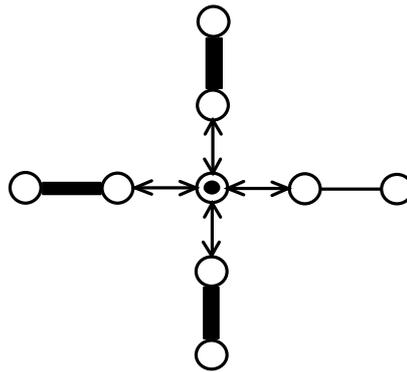


Figura 10. Informação de vizinhança extraída do Autômato de Mapeamento: três locais ocupados, quatro locais livres e um local ainda não mapeado.

A extração desta informação é feita após a conclusão de cada operação do Autômato de Mapeamento, na qual este autômato registra, entre estados pré-determinados, a informação contida nos estados vizinhos à posição corrente. O Sistema Gerente consulta tais estados pré-determinados para a obtenção da informação.

#### 5.4 Descrição do Autômato de Mapeamento

O Autômato de Mapeamento proposto recebe, como cadeia de entrada, informação do conjunto de sensores e informação de movimento previamente definida pelo Autômato de Exploração, e sua operação deve permitir registrar estes dados coletados pelos sensores de forma coerente, relacionando-os com o mapa já adquirido. A informação de entrada é

codificada pelo Sistema Gerente para poder ser processada pelo Autômato de Mapeamento. Tal codificação é descrita na tabela 1.

Tabela 1 – Codificação da cadeia de entrada do Autômato de Mapeamento.

<b>Codificação</b>	<b>Significado da codificação</b>	<b>Procedência da informação</b>
a	Norte livre	Sistema sensor
b	Sul livre	Sistema sensor
c	Leste livre	Sistema sensor
d	Oeste livre	Sistema sensor
h	Norte ocupado	Sistema sensor
i	Sul ocupado	Sistema sensor
j	Leste ocupado	Sistema sensor
k	Oeste ocupado	Sistema sensor
n	Deslocamento em direção norte	Autômato de Exploração
s	Deslocamento em direção sul	Autômato de Exploração
l	Deslocamento em direção leste	Autômato de Exploração
o	Deslocamento em direção oeste	Autômato de Exploração
r	Registrar marco na posição corrente	Autômato de Exploração
m	Fim da cadeia	-

Além de ser codificada, a informação de entrada também deve figurar em uma ordem pré-estabelecida. O Sistema Gerente também é responsável por esta ordenação, que é descrita na tabela 2.

A representação interna das informações coletadas pelos sensores sobre a topologia do ambiente também ocorre segundo uma codificação definida. Esta codificação contempla ainda outras informações de apoio utilizadas pelo autômato e está descrita na tabela 3.

Tabela 2. Ordem dos caracteres da cadeia de entrada do Autômato de Mapeamento.

1	2	3	4	5	6	'm'
1	Condição da direção norte					
2	Condição da direção sul					
3	Condição da direção leste					
4	Condição da direção oeste					
5	Indicação da direção do próximo deslocamento realizado pelo robô					
6	Marco definido na posição corrente (caractere opcional)					

Nota-se que as quatro primeiras linhas da tabela 3 apresentam dois caracteres codificadores para representar áreas não mapeadas. Este fato ocorre porque se utilizam duas transições para tais representações.

Como dado de saída, o Autômato de Mapeamento compõe a informação de vizinhança, que é coletada e enviada ao Autômato de Exploração pelo Sistema Gerente. Tal composição é realizada pela inclusão de transições contendo esta informação entre estados pré-determinados. A figura 11 contém um exemplo da representação gráfica da saída que indica condição norte mapeada como livre para deslocamento.

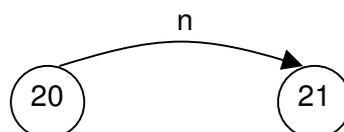


Figura 11. Representação gráfica da informação norte de saída do Autômato de Mapeamento.

Esta saída também é feita de acordo com uma codificação definida, a qual está apresentada na tabela 5, e os estados pré-definidos como portadores das transições de saída que compõem a informação de vizinhança estão relacionados na tabela 4.

Tabela 3 – Codificação interna do Autômato de Mapeamento.

<b>Codificação</b>	<b>Significado da codificação</b>
a / h	Norte não mapeado
b / i	Sul não mapeado
c / j	Leste não mapeado
d / k	Oeste não mapeado
n	Norte mapeado como condição livre
s	Sul mapeado como condição livre
l	Leste mapeado como condição livre
o	Oeste mapeado como condição livre
w	Norte mapeado como condição ocupada
x	Sul mapeado como condição ocupada
y	Leste mapeado como condição ocupada
z	Oeste mapeado como condição ocupada
f	Identificador de vértice noroeste
g	Identificador de vértice sudeste
p	Identificador de vértice nordeste
q	Identificador de vértice sudoeste

Tabela 4 – Estados do Autômato de Mapeamento que contêm as transições com informação de saída.

<b>Estados</b>	<b>Informação de saída</b>
20 e 21	Norte-adjacente
22 e 23	Norte-distante
24 e 25	Sul-adjacente
26 e 27	Sul-distante
28 e 29	Leste-adjacente
30 e 31	Leste-distante
31 e 32	Oeste-adjacente
33 e 34	Oeste-distante

A seguir são descritas, de forma detalhada, como as ações adaptativas permitem a execução das diversas operações que devem ser realizadas pelo Autômato de Mapeamento. Tais operações são:

1. Modificação da estrutura do autômato que está diretamente conectada com o estado corrente segundo a informação de entrada fornecida pelo Sistema Gerente;
2. Registro dos estados especiais que forem identificados como marcos do ambiente;
3. Registro do estado corrente do autômato a fim de conservar informação sobre o posicionamento do robô para a próxima execução do autômato;
4. Construção da informação de saída do Autômato de Mapeamento (conforme figura 10);
5. Expandir a estrutura do autômato, segundo movimento executado pelo robô, após a verificação da necessidade de se fazê-lo (de acordo com a proximidade do estado corrente com a borda da estrutura do autômato, conforme descrito no item 5.3).

Os nomes que identificam as ações adaptativas, responsáveis pelo comportamento do autômato, seguem a ordem alfabética, salvo exceções que ocorreram devido ao processo de elaboração do trabalho.

Tabela 5 – Codificação de saída do Autômato de Mapeamento.

<b>Codificação</b>	<b>Significado da codificação</b>
a	Norte não mapeado
b	Sul não mapeado
c	Leste não mapeado
d	Oeste não mapeado
n	Norte mapeado como condição livre
s	Sul mapeado como condição livre
l	Leste mapeado como condição livre
o	Oeste mapeado como condição livre
w	Norte mapeado como condição ocupada
x	Sul mapeado como condição ocupada
y	Leste mapeado como condição ocupada
z	Oeste mapeado como condição ocupada

A configuração inicial do Autômato de Mapeamento é composta por padrões de estados e transições que se repetem com a finalidade de compor o formato de treliça, conforme apresentado na figura 5. A figura 12 mostra a representação gráfica do padrão de estados e transições que compõem o canto superior esquerdo desta treliça.

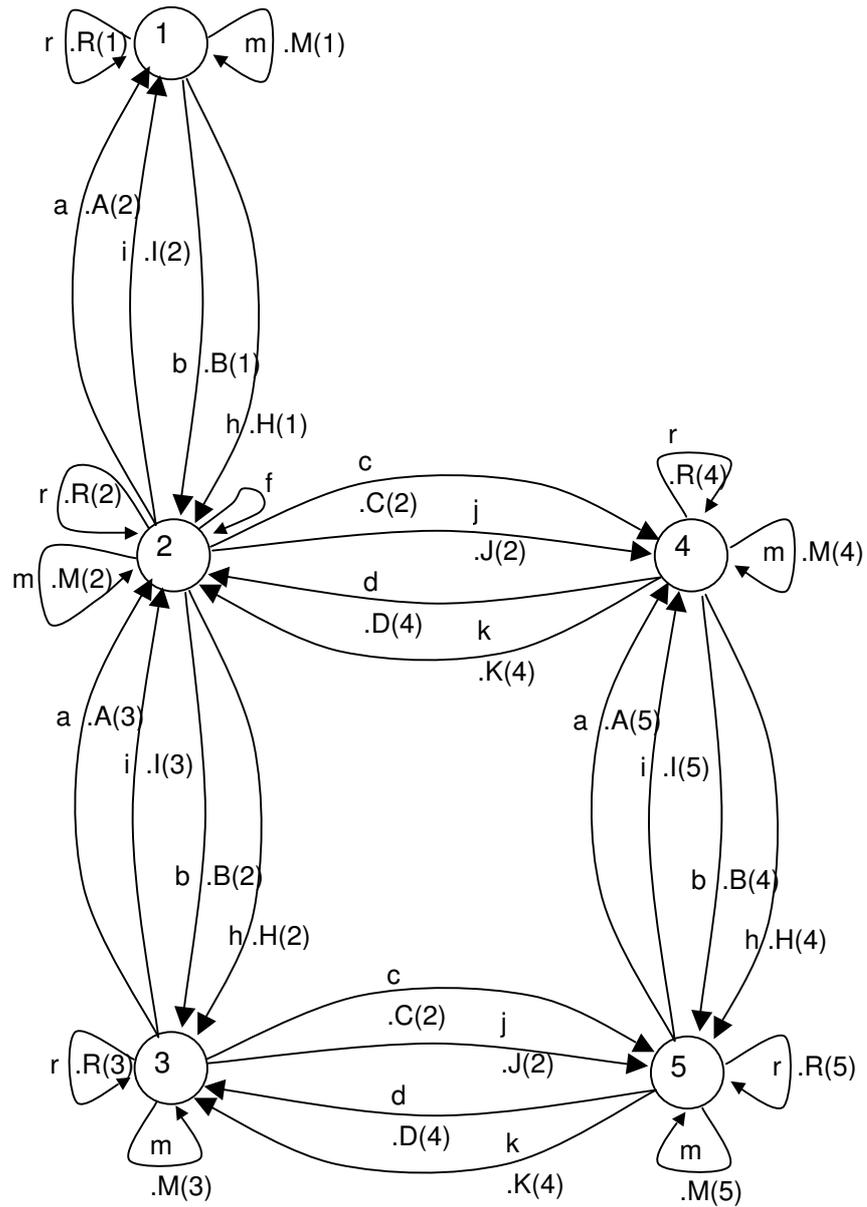


Figura 12. Representação gráfica padrão de transições que compõem o canto superior esquerdo Autômato de Mapeamento.

Na figura 12, supondo o estado 5 como o estado corrente do autômato, ao se processar o caractere 'a' de entrada, o qual significa que os sensores detectaram que o espaço ao norte está livre, executa-se a ação adaptativa A com parâmetro que indica o estado de origem da transição (5). A ação adaptativa A é descrita na figura 13.

```

A(e) {var1:
-[(e,a):A->var1]
-[(e,h):H->var1]
-[(var1,b):B->e]
-[(var1,i):I->e]

+[(var1,b)->var1]
+[(e,n):NN->var1:FF]
+[(var1,s):SS->e:GG]
+[(e,a)->e]}

```

Figura 13. Ação adaptativa A do Autômato de Mapeamento.

A ação adaptativa A substitui o conjunto de transições que interconectam os estados 4 e 5, indicativas de local ainda não explorado, por transições que indicam local explorado e livre para deslocamento (caracteres 'n' e 's'). As transições com estes dois caracteres têm o objetivo de permitir a mudança de estado de acordo com a informação de movimento que será indicada na cadeia de entrada do Autômato de Mapeamento. A ação adaptativa A insere ainda uma transição reflexiva para o consumo do caractere 'a', que está sendo processado, sem que ocorra uma mudança de estado. Esta transição permite que seja realizado novamente o processamento do caractere 'a', caso a exploração do ambiente conduza o robô a passar novamente pelo mesmo local, sem que seja executada novamente a ação adaptativa de substituição de transições (processo semelhante é realizado com o caractere 'b'). A figura 14 mostra a configuração do Autômato de Mapeamento após a conclusão ação adaptativa A. Existem outras ações adaptativas, similares à ação adaptativa A, responsáveis pelo processamento das informações coletadas pelos sensores indicando condição livre em outras direções. Estas ações estão relacionadas na tabela 6.

Semelhante às ações adaptativas responsáveis pelo registro de informações sobre condições livres no Autômato de Mapeamento, têm-se também ações adaptativas responsáveis pelo registro de informações sobre condições ocupadas. Estas ações substituem o grupo de

transições que indicam locais não explorados por transições que indicam locais obstruídos para o deslocamento e também adicionam transições reflexivas ao estado em questão de modo que, caso o robô percorra novamente o mesmo local durante sua exploração, as informações coletadas pelo sistema sensor possam ser processadas sem que haja necessidade de nova substituição de transições. As ações estão relacionadas na tabela 7.

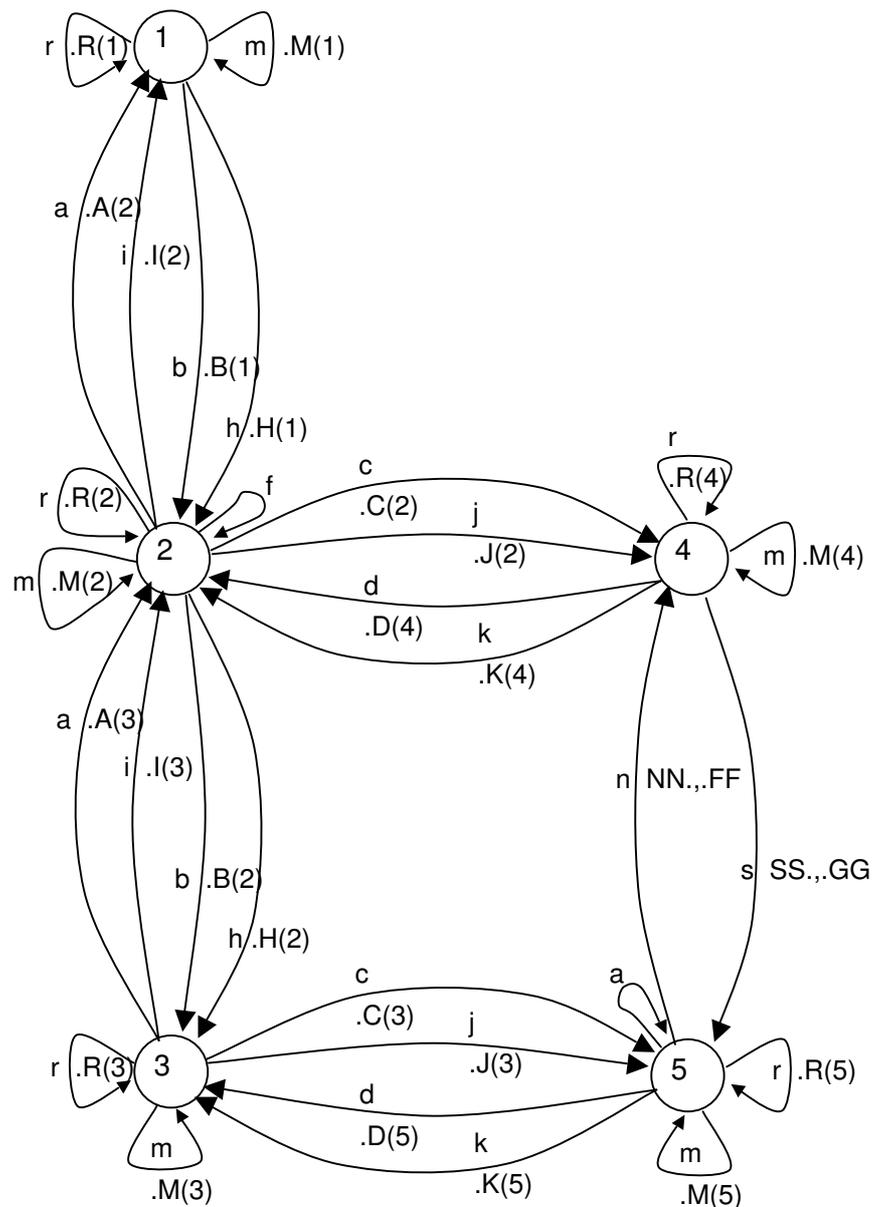


Figura 14. Representação gráfica padrão de transições que compõem o canto superior esquerdo Autômato de Mapeamento após a execução da primeira ação adaptativa (A).

Tabela 6 – Ações adaptativas semelhantes responsáveis pelo registro de condição livre.

<b>Ação adaptativa</b>	<b>Função</b>
A	Registro de condição livre na direção norte
B	Registro de condição livre na direção sul
C	Registro de condição livre na direção leste
D	Registro de condição livre na direção oeste

Tabela 7 – Ações adaptativas semelhantes responsáveis pelo registro de condição ocupada.

<b>Ação adaptativa</b>	<b>Função</b>
H	Registro de condição ocupada na direção norte
I	Registro de condição ocupada na direção sul
J	Registro de condição ocupada na direção leste
K	Registro de condição ocupada na direção oeste

Nota-se nas figuras 12 e 14 a ocorrência, em todos os estados, da transição reflexiva que consome o caractere 'r' optativo na cadeia de entrada. A existência deste caractere na cadeia de entrada do autômato indica que o estado corrente deve ser definido como um estado especial. Este estado é chamado de marco e assinala um ponto de apoio para a exploração do ambiente. A figura 15 apresenta a ação adaptativa R que executa esta demarcação criando uma transição entre o estado especial 45 e o estado corrente.

$$R(e) \{ : \\ + [ ( 45 , r ) -> e ] \}$$

Figura 15. Ação adaptativa R do Autômato de Mapeamento.

Ainda nas figuras 12 e 14, ocorre também em todos os estados, a transição reflexiva que consome o caractere 'm' indicador de final da cadeia de entrada e executa a ação adaptativa M apresentada na figura 16.

```

M(e) {var1:
-[(44,#):V->var1]
+[(44,#):V->e]
-[(23,a)->22]
-[(27,a)->26]
-[(31,a)->30]
-[(35,a)->34]

T(e,e,e,e)}

```

Figura 16. Ação adaptativa M do Autômato de Mapeamento.

A ação adaptativa M inicialmente move a transição que consome o símbolo vazio<sup>2</sup> para o estado corrente, registrando o posicionamento atual para a próxima execução do Autômato Adaptativo e, em seguida, apaga as transições de apoio indicativas de saída para que seja possível iniciar a construção das novas transições com as informações de saída. Para a construção da saída indicativa da condição norte, a ação adaptativa final T é chamada. A ação adaptativa, descrita na figura 17, utiliza o estado 15 apenas como apoio para inserir, nos estados pré-determinados, a codificação da informação de saída apresentada na tabela 5.

```

T(e,ee,uu,u)
{var1,var2,var3,var4,var5:
?[(e,a):A->var1]
?[(ee,n):NN->var2:FF]
?[(uu,w)->var3]

+[(15,a)->var1]
+[(15,n)->var2]
+[(15,w)->var3]

?[(15,var4)->var5]
+[(20,var4)->21]
-[(15,var4)->var5]

TT(var5,var5,var5,u)}

```

Figura 17. Ação adaptativa T do Autômato de Mapeamento.

<sup>2</sup> O símbolo vazio  $\epsilon$  é representado nas ações adaptativas em todo o texto com caractere especial #.

Existem outras ações adaptativas, semelhantes à ação adaptativa T, que são responsáveis pelas construções das demais saídas existentes e estão descritas na tabela 8. Estas ações adaptativas estão encadeadas de forma que cada ação chame a próxima, executando-se para tal, sucessivas chamadas de ações adaptativas finais.

Tabela 8 – Ações adaptativas responsáveis pela construção da saída do Autômato de Mapeamento.

<b>Ação adaptativa</b>	<b>Saída executada</b>
T	Norte adjacente
TT	Norte distante
U	Sul adjacente
UU	Sul distante
X	Leste adjacente
XX	Leste distante
Z	Oeste adjacente
ZZ	Oeste distante

O estado 44, que é a origem da transição que consome o caractere vazio movida pela ação adaptativa M (figura 16), é o estado final de uma seqüência de estados conectados ao estado inicial do Autômato de Mapeamento (estado 40). Tal seqüência é apresentada na figura 18.

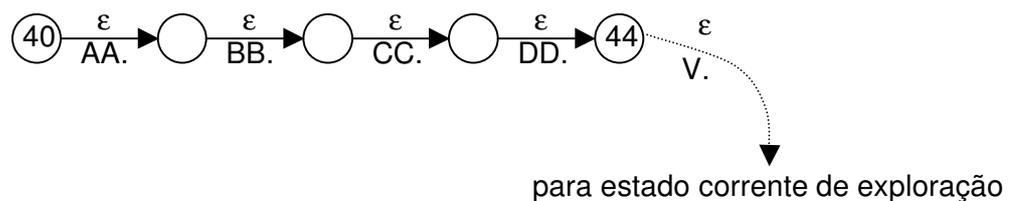


Figura 18. Seqüência de estados interligando o estado inicial ao estado corrente de exploração.

A seqüência de estados e transições que conectam o estado inicial do Autômato de Mapeamento ao estado corrente de exploração não processa nenhum caractere da cadeia de entrada, contudo tem a função de executar algumas ações adaptativas preliminares. Uma destas ações adaptativas (AA) é apresentada na figura 19. As demais ações semelhantes a esta estão relacionadas na tabela 9.

A transição que consome o símbolo vazio  $\epsilon$  com origem no estado 44 é movida para o estado atual durante execução do Autômato de Mapeamento com a finalidade de que, quando este autômato for novamente acionado, seu estado inicial esteja direcionado novamente para o estado corrente de exploração.

AA{var1: ?[(22,var1)->23] +[(23,a)->22]  HH(var1)}	HH( $\epsilon$ ){: -[(23,a)->22]}
--	--------------------------------------

Figura 19. Ações adaptativas AA e HH do Autômato de Mapeamento.

A ação adaptativa AA e sua ação adaptativa final HH têm a função de sinalizar se o Autômato de Mapeamento deve ou não ser ampliado na direção norte, caso o deslocamento de exploração seja na direção norte. Esta sinalização é feita através da inclusão de uma transição que consome o caractere 'a' entre os estados 23 e 22 e é condicionada a uma consulta da condição da informação registrada na saída entre estes dois estados (saída norte-distante). Caso não haja informação registrada entre os estados 22 e 23 esta transição é adicionada, significando a possibilidade de ampliação da estrutura e, em caso contrário, a transição é apagada.

Tabela 09 – Ações adaptativas iniciais do Autômato de Mapeamento.

<b>Ação adaptativa</b>	<b>Função</b>
AA	Sinalizar a possibilidade de ampliação do autômato na direção norte
BB	Sinalizar a possibilidade de ampliação do autômato na direção sul
CC	Sinalizar a possibilidade de ampliação do autômato na direção leste
DD	Sinalizar a possibilidade de ampliação do autômato na direção oeste

Tem-se também, na seqüência de estados da figura 18, a ação adaptativa V que é executada no processamento da última transição com o símbolo vazio  $\epsilon$ . Esta ação adaptativa, executada no início de cada ciclo de operação do Autômato de Mapeamento, tem a função de apagar as informações de saída deste autômato registradas no ciclo de operação anterior. Tal ação adaptativa V está apresentada na figura 20.

```
V{var1, var2, var3, var4, var5, var6, var7, var8 :
-[(20, var1)->21]
-[(22, var2)->23]
-[(24, var3)->25]
-[(26, var4)->27]
-[(28, var5)->29]
-[(30, var6)->31]
-[(32, var7)->33]
-[(34, var8)->35]}
```

Figura 20. Ação adaptativa V do Autômato de Mapeamento.

Finalmente, na figura 13, supondo o estado 5 como estado corrente do autômato e o caractere da cadeia de entrada indicativo de deslocamento norte ('n'), haverá uma mudança para o estado 3 e as regras adaptativas NN e FF serão executadas. Estas ações adaptativas executam a expansão do Autômato de Mapeamento e estão descritas nas figuras 21 e 22, respectivamente.

<pre> NN{var1,var2: ?[(var1,f)-&gt;var1] ?[(23,var2)-&gt;22]  N(var1,var2)} </pre>	<pre> N(e,ee){var1,var2,ger1*: ?[(e,a):A-&gt;var1] ?[(e,c):C-&gt;var2]  +[(ger1,b):B(ger1)-&gt;var2] +[(ger1,i):I(ger1)-&gt;var2] +[(var2,a):A(var2)-&gt;ger1] +[(var2,h):H(var2)-&gt;ger1] +[(var1,c):C(var1)-&gt;ger1] +[(var1,j):J(var1)-&gt;ger1] +[(ger1,d):D(ger1)-&gt;var1] +[(ger1,k):K(ger1)-&gt;var1]  +[(ger1,r):R(ger1)-&gt;ger1] +[(ger1,m):M(ger1)-&gt;ger1]  N(var2,ee)} </pre>
--	--

Figura 21. Ações adaptativas NN e N do Autômato de Mapeamento.

A ação adaptativa NN verifica a sinalização de ampliação do autômato entre os estados 23 e 22 e, caso haja tal indicação, chama a ação adaptativa final N com parâmetro que indica o vértice a partir do qual o autômato deverá ser ampliado.

A ação adaptativa N opera recursivamente, adicionando, a cada execução, um novo estado (e as transições que o conectam) à parte superior do autômato. A recursividade da operação desta ação adaptativa é interrompida quando a linha a que os novos estados se conectam chegar ao final.

Executada também apenas quando uma ampliação do autômato é realizada, a ação adaptativa F tem a função de inserir novas transições especiais de suporte às futuras expansões da estrutura, apagar as transições de suporte antigas e mover as transições reflexivas indicativas das arestas do autômato, concluindo assim o processo de crescimento do Autômato de Mapeamento. As tabelas 10 e 11 apresentam as ações adaptativas similares, respectivamente, às ações NN e FF.

<pre> FF{var1: ?[(23,var1)-&gt;22]  F(var1)} </pre>	<pre> F(e){var1,var2,var3,var4,var5,var6, var7,var8,var9,var10,var11,var12, var13,var14,var15,var16,ger1*:  -[(var16,f)-&gt;var16] ?[(var16,a):A-&gt;var1] +[(var1,f)-&gt;var1] +[(ger1,b):B(ger1)-&gt;var1] +[(ger1,i):I(ger1)-&gt;var1] +[(var1,a):A(var1)-&gt;ger1] +[(var1,h):H(var1)-&gt;ger1] +[(ger1,r):R(ger1)-&gt;ger1] +[(ger1,m):M(ger1)-&gt;ger1]  ?[(var4,p)-&gt;var4] ?[(var4,c):C-&gt;var5] ?[(var4,a):A-&gt;var6]  -[(var4,c):C-&gt;var5] -[(var4,j):J-&gt;var5] -[(var5,d):D-&gt;var4] -[(var5,k):K-&gt;var4]  -[(var7,b):B-&gt;var5] -[(var7,i):I-&gt;var5] -[(var5,a):A-&gt;var7] -[(var5,h):H-&gt;var7]  -[(var7,c):C-&gt;var8] -[(var7,j):J-&gt;var8] -[(var8,d):D-&gt;var7] -[(var8,k):K-&gt;var7]  -[(var5,r):R-&gt;var5] -[(var8,r):R-&gt;var8] -[(var5,m):M-&gt;var5] -[(var8,m):M-&gt;var8]  -[(var4,p)-&gt;var4] +[(var6,p)-&gt;var6]} </pre>
---	--

Figura 22. Ações adaptativas FF e F do Autômato de Mapeamento.

Tabela 10 – Ações adaptativas semelhantes responsáveis pela expansão do Autômato de Mapeamento.

<b>Ação adaptativa</b>	<b>Função</b>
NN	Ampliação do autômato através da adição de uma linha à parte norte
SS	Ampliação do autômato através da adição de uma linha à parte sul
LL	Ampliação do autômato através da adição de uma coluna à parte leste
OO	Ampliação do autômato através da adição de uma coluna à parte oeste

Tabela 11 – Ações adaptativas semelhantes responsáveis pela conclusão do processo de ampliação do autômato.

<b>Ação adaptativa</b>	<b>Função</b>
FF	Conclusão do processo de expansão no autômato na direção norte
GG	Conclusão do processo de expansão no autômato na direção sul
PP	Conclusão do processo de expansão no autômato na direção leste
QQ	Conclusão do processo de expansão no autômato na direção oeste

### 5.5 Considerações sobre o Autômato de Mapeamento

A figura 9 demonstra que a configuração final do Autômato de Mapeamento contém transições que denotam espaços ocupados, espaços livres e espaços não mapeados – na parte inferior direita desta figura – os quais são necessários para completar a configuração retangular do autômato. A proposta da manutenção da configuração retangular do Autômato de Mapeamento tem como objetivo correlacionar áreas distintas no espaço evitando a realização de cálculos posteriores que teriam o objetivo de determinar a relação entre duas áreas adjacentes no mapa que, devido à trajetória realizada pelo robô na exploração do ambiente, foram mapeadas em momentos diferentes.

Observando-se as figuras 5, 8 e 9 nota-se que a geração de estados e transições conectando esses estados no Autômato de Mapeamento acontece de acordo com a movimentação do robô pelo ambiente. Os estados e transições registram as informações coletadas pelo conjunto de sensores do robô e são responsáveis pela ocupação de memória do sistema apresentado. Tal característica da estrutura de registro proposta implica que o aumento da utilização de memória do sistema proposto aconteça de acordo com ambiente físico já explorado.

Dentro da divisão da área de pesquisa de mapeamento robótico em abordagens métricas e topológicas, apresentadas no capítulo 2, pode-se classificar o presente trabalho como um método híbrido, pois contém características de ambas as abordagens. A estrutura de mapeamento modela as características geométricas do ambiente e as dimensões dos obstáculos presentes neste ambiente estão implícitas no modelo. Tal propriedade é peculiar às abordagens métricas de mapeamento. Porém, o mapa final construído pelo autômato contém informações de deslocamento entre pontos diferentes do espaço. Esta propriedade é característica das estratégias topológicas de mapeamento.

## **6. AUTÔMATO DE EXPLORAÇÃO**

Como descrito no capítulo 3, o presente trabalho emprega os Autômatos Adaptativos na definição da trajetória do robô durante a exploração de ambientes desconhecidos. Este capítulo apresenta o comportamento do Autômato de Exploração durante a exploração de um ambiente desconhecido no item 6.2, as características do autômato que permitem o desempenho deste comportamento no item 6.3 e as considerações sobre o autômato no item 6.4. Para tanto, apresentam-se a seguir os princípios adotados para a definição do seu comportamento comparados com as características de outros modelos existentes. O Autômato de Exploração foi apresentado no trabalho de Sousa, Hirakawa e Neto (2004a) e o Apêndice B contém sua descrição algébrica completa, de acordo com a notação apresentada no capítulo 4 sugerida na pesquisa de Neto (1993), exceto pela representação do símbolo vazio  $\epsilon$  com o caractere #.

### **6.1 Introdução**

Pode-se entender o processo de exploração de locais desconhecidos como o conjunto de operações e deslocamentos realizados pelo agente robótico com a finalidade de construir uma representação abstrata do ambiente no qual está inserido. O trabalho de Batalin e Sukhatme (2003) apresenta a exploração de ambientes por robôs classificada em dois modelos: (i) estático, no qual os sensores do robô podem captar informações sobre todo o ambiente a todo instante, e (ii) dinâmico, no qual os sensores do robô não captam informações de todo ambiente, e sim, necessitam que o robô se desloque pelo ambiente para que o reconhecimento do local seja realizado. A presente proposta pertence ao segundo modelo, pois, por não se

restringir a nenhum tipo específico de sensores, adota-se que tais sensores podem não obter informações sobre o completo ambiente sem a necessidade de deslocamento do robô.

As abordagens iniciais da questão da exploração de ambientes não empregavam estratégias bem definidas para a realização deste processo e a aquisição de terreno era feita de maneira acidental, durante a navegação do robô por áreas já conhecidas (RAO et al., 1987) ou pelo uso de técnicas puramente reativas (YAMAUCHI, 1997). Tais abordagens não obtinham a eficiência necessária e, com a evolução desta área de pesquisa, desenvolveram-se dois tipos de estratégias diferentes para a exploração de ambientes. O primeiro método tem como objetivo visitar todos os obstáculos (ou vértices de obstáculos) presentes no local onde o robô se encontra e, para aquisição completa do mapa do ambiente, assume que todos os obstáculos presentes no ambiente são mutuamente visíveis. O segundo método tem como fundamento permitir que o robô percorra todos os pontos do local a ser explorado com a finalidade de obter o completo mapa do ambiente em questão (LUMELSKY; MUKHOPADHYAY; SUN, 1990). Como a proposta apresentada neste trabalho é aplicada em qualquer tipo de ambiente no qual o robô esteja inserido, o sistema de exploração escolhido pertence ao segundo grupo de métodos descrito.

O processo que permite ao robô visitar todos os pontos do espaço em que está inserido é chamado de cobertura completa de ambientes e, além de ser empregado no mapeamento robótico, também é aplicado em tarefas de limpeza, detecção de minas terrestres e aparagem de grama desempenhada por agentes robóticos (YASUTOMI, 1988; CARVALHO et al., 1997). A estratégia de cobertura completa de ambientes empregada no presente trabalho utiliza uma trajetória em ziguezague. Esta abordagem é comum a diversas aplicações, como por exemplo, o trabalho apresentado por Schmidt e Hofner (1998), que desenvolve a estratégia de cobertura de ambientes em ziguezague com o objetivo de limpeza ou inspeção de áreas industriais e públicas, porém – diferente do proposto no presente trabalho – utiliza

mapas já construídos para a execução da tarefa e elabora trajetórias alternativas caso o robô se depare com algum obstáculo móvel não previamente mapeado; a pesquisa de Luo, Yang e Meng (2002) possibilita que o robô percorra em ziguezague todos os pontos do ambiente no qual está inserido através da utilização de uma rede neural. Cada neurônio da rede neural é relacionado a uma porção do espaço em questão. Áreas já visitadas e áreas que contêm obstáculos adquirem pesos que repelem o robô e áreas ainda não visitadas, pesos que atraem o robô. Entretanto, a estrutura apresentada – diferente da proposta deste trabalho – necessita que o robô tenha conhecimento prévio das dimensões da área a ser preenchida; e o trabalho de Acar e Choset (2001), que também adota uma trajetória em ziguezague para a exploração completa de um ambiente em que o robô se encontra e exemplifica a complexidade dos algoritmos envolvidos na execução desta tarefa, sobretudo quando a finalidade é a cobertura inteira de um local sobre o qual não se dispõe de nenhuma informação *a priori* de suas dimensões ou topologia – situação análoga à proposta por este trabalho.

Além da estratégia que emprega trajetórias em ziguezague para completa cobertura de ambientes, outra técnica comum emprega trajetórias em espiral para a execução da mesma tarefa, entretanto, tais trajetórias não exibem uma eficiência adequada em ambientes constituídos por configurações complexas (FU; LANG, 1999).

A trajetória em ziguezague também foi escolhida para o presente trabalho pela sua generalidade e porque o princípio adotado exige que o robô não tenha nenhuma informação prévia do ambiente a ser mapeado. Algumas abordagens de exploração de ambientes dividem o espaço no qual o robô está inserido em retângulos e esquadriham estas divisões, como, por exemplo, o trabalho de González et al. (1996). Nota-se, entretanto, que uma trajetória em ziguezague pode ser interpretada como um retângulo de lado unitário, cujo interior é completamente conhecido, o que é uma característica importante para a aplicação de mapeamento.

De acordo com a estratégia de exploração de ambientes já exposta, utiliza-se, como técnica para detecção de locais ainda não explorados, um procedimento similar ao utilizado na pesquisa de Yamauchi (1997), na qual é proposto um algoritmo que conduz o robô a localizar – e se mover para – uma fronteira entre um lugar desconhecido e outro lugar conhecido, já mapeado e acessível. Esta técnica é implementada no Autômato de Exploração com a busca, durante o retorno de um local já completamente explorado, pelo conjunto de transições adjacentes “livre – não mapeado” na informação extraída do Autômato de Mapeamento pelo Sistema Gerente. Esta seqüência de transições (chamada de informação de vizinhança) indica se existem áreas ainda não mapeadas pelo robô e se o caminho para acessar estas áreas está livre.

Segundo o algoritmo de exploração apresentado no item anterior, quando o Autômato de Mapeamento encontra uma região ainda não mapeada do ambiente, ele conduz o robô a explorá-la completamente antes de buscar por outra região ainda não mapeada. Este tipo busca, que procura esgotar inteiramente o caminho corrente, é comum a outros trabalhos já desenvolvidos, como por exemplo, na pesquisa de FU e LANG (1999), e foi escolhida porque, como o acesso de áreas ainda não mapeadas é feito através do deslocamento do robô pelo ambiente, é mais econômico para o processo esgotar completamente a área que está sendo explorada antes de se mover para outra desconhecida. Esta abordagem permite ainda a utilização do autômato proposto na condução do robô durante a execução de tarefas em que o objetivo não é conhecer todo o ambiente, e sim, dado qualquer posição inicial de deslocamento, o robô encontrar uma posição final, aleatoriamente escolhida, dentro de um ambiente sobre o qual este robô não possui nenhuma informação *a priori*.

## 6.2 Comportamento do Autômato de Exploração

A operação do Autômato de Exploração permite ao agente robótico explorar completamente o ambiente no qual está inserido descrevendo uma trajetória em ziguezague de acordo com o seguinte algoritmo:

1. Iniciando em qualquer ponto do ambiente, o autômato conduz o robô para a direção norte até encontrar um obstáculo. A figura 23 exemplifica este deslocamento.

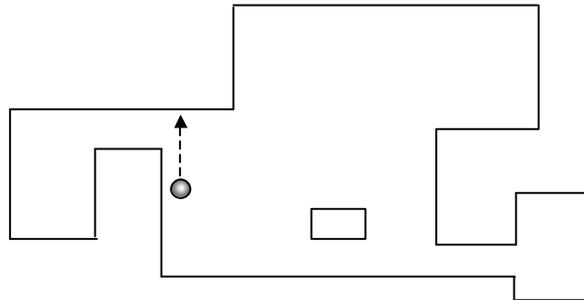


Figura 23. Movimento de exploração inicial em direção ao norte.

2. O autômato inicia a condução do robô segundo a trajetória em ziguezague – do norte para o sul – retornando pelo caminho paralelo – do sul para o norte – assim que encontrar um obstáculo. Esta trajetória em ziguezague cresce na direção leste. A figura 24 exemplifica este deslocamento.
3. Após cada deslocamento para o leste, e antes da continuação da trajetória em ziguezague, o autômato confere – pela leitura dos sensores – a existência de áreas livres na direção oposta ao próximo deslocamento, preenchendo este espaço até encontrar um obstáculo, retomando em seqüência a trajetória em ziguezague. Esta ação é chamada de movimento complementar de deslocamento leste e é exemplificado na figura 25.

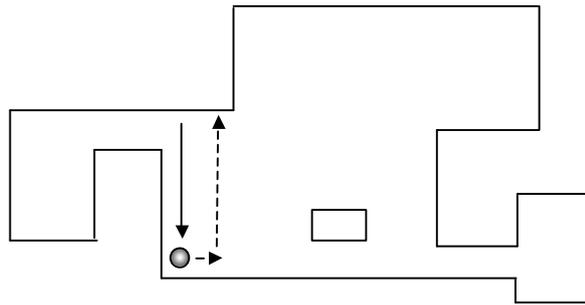


Figura 24. Início da trajetória em ziguezague.

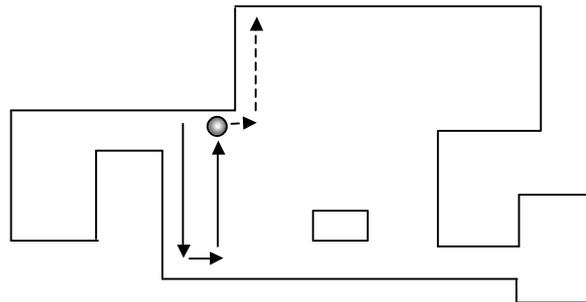


Figura 25. Preenchimento de área livre localizada na direção oposta ao movimento em ziguezague.

4. Tal procedimento é repetido até a trajetória em ziguezague ser esgotada. O encerramento da trajetória em ziguezague pode ser dado pelo término da porção do espaço que está sendo explorado ou pelo encontro de uma área já mapeada anteriormente pelo robô. A figura 26 exemplifica o esgotamento da trajetória em ziguezague pelo término do espaço.

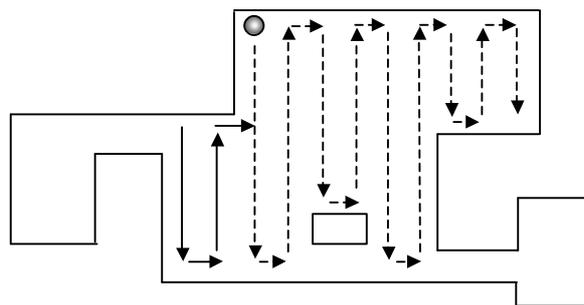


Figura 26. Exaustão do processo de cobertura de área.

5. Após o esgotamento da trajetória realizada pelo agente robótico, o autômato conduz o robô de forma que este retorne pela mesma trajetória buscando por uma seqüência de transições adjacentes “livre – não mapeado” localizadas nas direções leste ou oeste. Para

tal propósito o Autômato de Exploração utiliza a informação extraída do Autômato de Mapeamento pelo Sistema Gerente exemplificada pela figura 10. Esta seqüência de transições indica que existe uma área ainda não mapeada pelo robô e que o caminho para acessar a área está livre. Ao detectar tal configuração, o autômato conduz o robô a explorar este novo espaço segundo a trajetória em ziguezague conforme descrito nos itens 1 a 4. Nota-se que o novo espaço pode ser um local simples ou um ambiente tão complexo quanto o ambiente já explorado. As figuras 27 e 28 exemplificam, respectivamente, a localização de ambientes não mapeados, e sua exploração, nas direções leste e oeste.

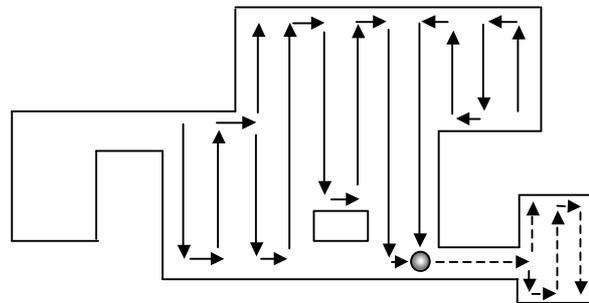


Figura 27. Exploração de área localizada a leste do robô.

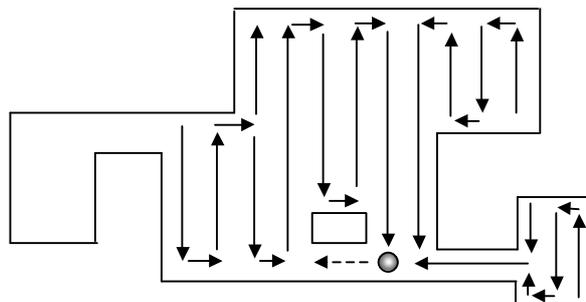


Figura 28. Exploração de área localizada a oeste do robô.

6. O procedimento é repetido até que o robô se encontre novamente em sua posição inicial de exploração. Em seguida, o procedimento é repetido substituindo-se a trajetória em ziguezague crescente para o leste pela trajetória crescente para o oeste. A figura 29 exemplifica este deslocamento.

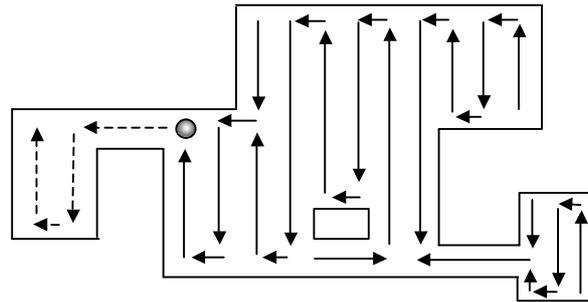


Figura 29. Exploração do ambiente segundo trajetória em ziguezague crescente para o oeste.

7. Assim que o agente robótico retorna novamente à sua posição inicial, o Autômato de Exploração sinaliza ao Sistema Gerente que o ambiente foi completamente explorado e mapeado. A figura 30 exemplifica o término da exploração do ambiente em questão.

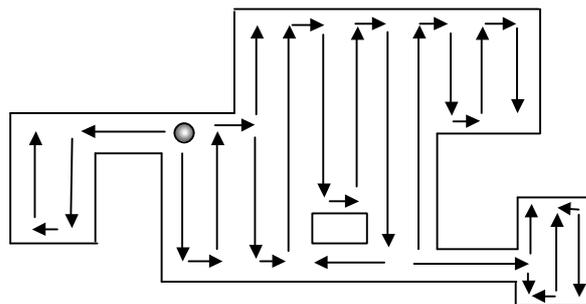


Figura 30. Término da exploração do ambiente.

### 6.2.1 Comportamento do Autômato de Exploração utilizando marcos

Conforme descrito no capítulo 3, durante o processo de exploração de ambientes o Autômato de Exploração sinaliza para o Autômato de Mapeamento alguns locais especiais, chamados de marcos. Estes marcos são definidos durante os movimentos norte e sul de exploração que registram e comparam a condição (livre ou ocupado) das regiões a leste e a oeste do robô. Se, no decurso de um deslocamento, for detectado alguma alteração na condição de ocupação de tais regiões o presente local é selecionado como marco. Por exemplo, durante o deslocamento

do robô seu conjunto de sensores detectou que havia uma região ocupada a leste e após uma movimentação para o norte esta região não está mais ocupada: este ponto é selecionado para ser demarcado. As oito possibilidades de determinação de marcos estão descritas nas próximas duas figuras, a figura 31 mostra a definição de marcos durante movimentos de regiões ocupadas para regiões livres e a figura 32 a situação inversa. O 'X' representa a posição do marco.

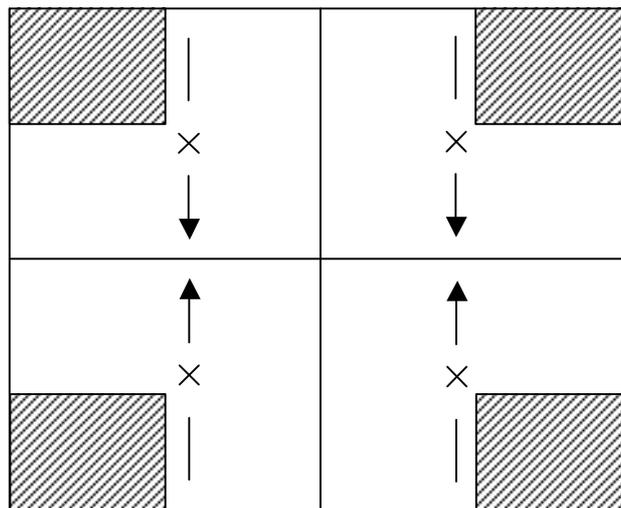


Figura 31. As quatro possibilidades de determinação de marcos durante o deslocamento de áreas ocupadas para áreas livres.

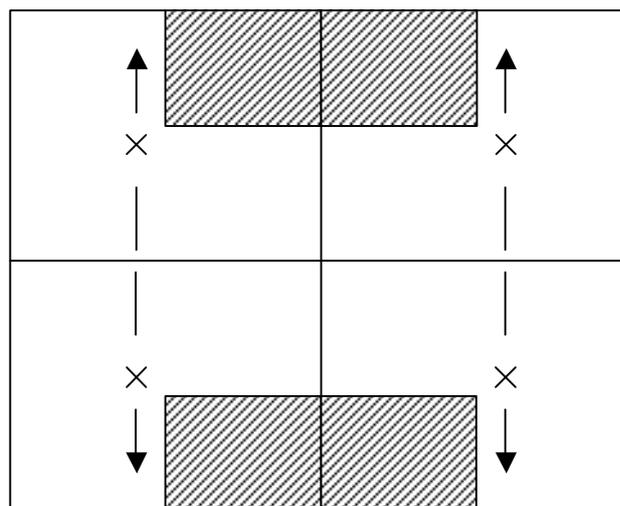


Figura 32. As quatro possibilidades de determinação de marcos durante deslocamento de áreas livres para áreas ocupadas.

Destaca-se que os marcos são definidos sempre nas áreas livres, ou seja, nas situações em que se detecta áreas livres e passa-se a detectar áreas ocupadas, como na figura 32, o marco deve ser determinado na posição em que o robô se encontrava anteriormente.

A definição de marcos para o ambiente explorado no exemplo proposto no item 6.2 está apresentada na figura 33.

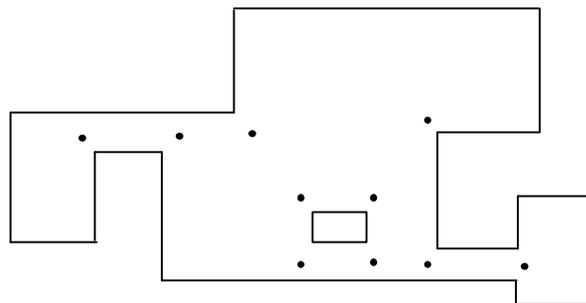


Figura 33. Marcos definidos durante a exploração do ambiente de exemplo.

A implementação da estratégia de marcos no Autômato de Exploração tem a função de evitar que o robô retorne pelo mesmo caminho de exploração para alcançar áreas ainda não exploradas. Como os marcos são definidos nas áreas entrecortadas do ambiente, eles sinalizam para qual ponto o robô deve retornar a fim de iniciar a exploração em uma área nova. As figuras 34, 35, 36 e 37 exemplificam como se dá o retorno de áreas já exploradas no exemplo apresentado no item 6.2 através do deslocamento de retorno entre marcos.

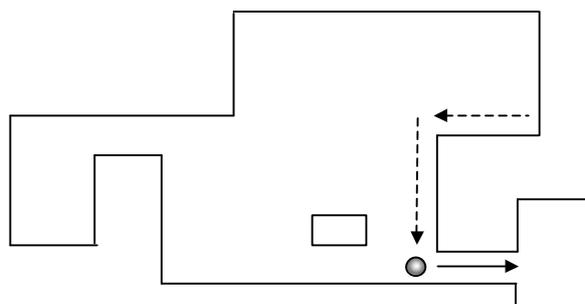


Figura 34. Retorno do 1º caminho esgotado em direção à 2ª área a ser explorada.

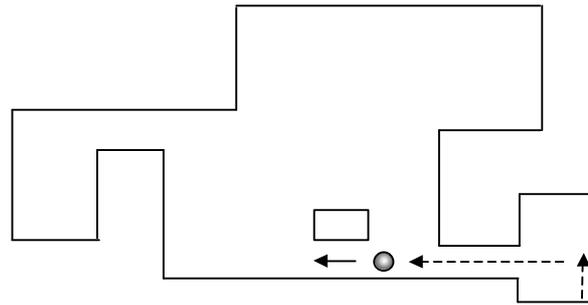


Figura 35. Retorno do 2º caminho esgotado em direção à 3ª área a ser explorada.

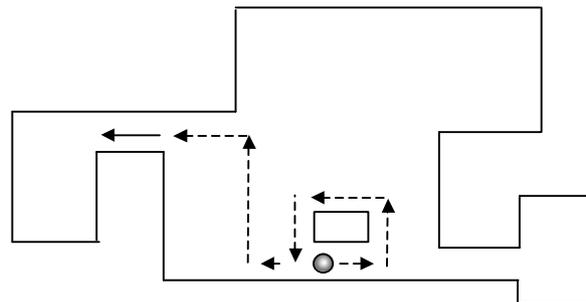


Figura 36. Retorno do 3º caminho esgotado em direção à última área a ser explorada.

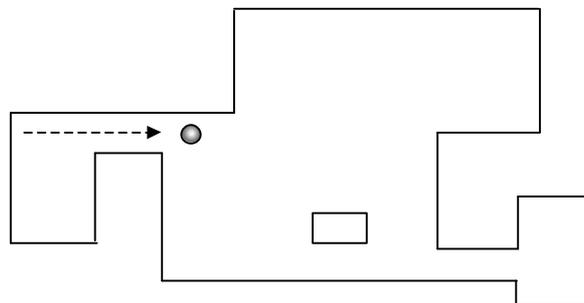


Figura 37. Retorno do último caminho esgotado em direção à posição inicial.

### 6.3 Configuração do Autômato de Exploração

O Autômato de Exploração proposto recebe, como cadeia de entrada, informação do conjunto de sensores e informação de vizinhança previamente registrada no Autômato de Mapeamento, e sua operação deve sinalizar qual a direção do próximo movimento a ser realizado pelo robô.

A informação de entrada é codificada pelo Sistema Gerente para ser processada pelo Autômato de Exploração, como descrita na tabela 12.

Tabela 12 – Codificação da cadeia de entrada do Autômato de Exploração.

<b>Codificação</b>	<b>Significado da codificação</b>	<b>Procedência da informação</b>
n	Norte livre	Mapa (informação de vizinhança)
s	Sul livre	Mapa (informação de vizinhança)
l	Leste livre	Mapa (informação de vizinhança)
o	Oeste livre	Mapa (informação de vizinhança)
w	Norte ocupado	Mapa (informação de vizinhança)
x	Sul ocupado	Mapa (informação de vizinhança)
y	Leste ocupado	Mapa (informação de vizinhança)
z	Oeste ocupado	Mapa (informação de vizinhança)
a	Norte não mapeado	Mapa (informação de vizinhança)
b	Sul não mapeado	Mapa (informação de vizinhança)
c	Leste não mapeado	Mapa (informação de vizinhança)
d	Oeste não mapeado	Mapa (informação de vizinhança)
e	Segmento não criado	Mapa (informação de vizinhança)
f	Norte livre	Conjunto de sensores
g	Sul livre	Conjunto de sensores
p	Leste livre	Conjunto de sensores
q	Oeste livre	Conjunto de sensores
m	Norte ocupado	Conjunto de sensores
t	Sul ocupado	Conjunto de sensores
u	Leste ocupado	Conjunto de sensores
v	Oeste ocupado	Conjunto de sensores
k	Fim da cadeia	-

Além de ser codificada, a informação de entrada também deve figurar em uma ordem pré-estabelecida. O Sistema Gerente também é responsável por esta ordenação, que está descrita na tabela 13.

Tabela 13. Ordem dos caracteres da cadeia de entrada do Autômato de Mapeamento.

1	2	3	4	5	6	7	8	9	10	11	12	'k'
1	Mapa (informação de vizinhança): norte-adjacente											
2	Mapa (informação de vizinhança): norte-distante											
3	Mapa (informação de vizinhança): sul-adjacente											
4	Mapa (informação de vizinhança): sul-distante											
5	Mapa (informação de vizinhança): leste-adjacente											
6	Mapa (informação de vizinhança): leste-distante											
7	Mapa (informação de vizinhança): oeste-adjacente											
8	Mapa (informação de vizinhança): oeste-distante											
9	Sensor: norte											
10	Sensor: sul											
11	Sensor: leste											
12	Sensor: oeste											

A sinalização de saída se dá pela inclusão de uma transição contendo a informação de movimento entre dois estados pré-determinados (implementado entre os estados 1 e 2), com a finalidade de permitir que o Sistema Gerente colete esta informação e componha a cadeia de entrada do Autômato de Mapeamento. A codificação da informação de saída está relacionada

na tabela 14 e um exemplo gráfico de saída indicando movimento norte é apresentada na figura 38.

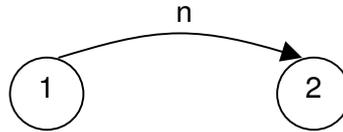


Figura 38. Representação gráfica da informação de saída do Autômato de Exploração.

Tabela 14. Codificação dos caracteres de saída do Autômato de Exploração.

<b>Codificação</b>	<b>Significado da codificação</b>
n	Movimento para o norte
s	Movimento para o sul
l	Movimento para o leste
o	Movimento para o oeste

Para tratar as informações de entrada, o autômato tem um conjunto de transições e estados que o permitem determinar a saída conforme os estímulos de entrada.

Durante a exploração de ambientes pode ocorrer o aparecimento de situações ambíguas, nas quais o autômato recebe informações de entrada similares e deve produzir saídas distintas de acordo com a configuração do ambiente e com o estágio corrente de exploração. Para tratar estas situações ambíguas existem divisões distintas no Autômato de Exploração, uma para cada estágio de exploração. As divisões podem ser conectadas ao estado inicial do autômato por uma transição e esta transição é direcionada para a divisão específica que deve tratar as informações dos sensores e de vizinhança de acordo com tipo de deslocamento que o robô está realizando. A tabela 15 apresenta todas as divisões do Autômato de Mapeamento com seus respectivos movimentos fundamentais e enumeradas pelos primeiros estados utilizados na implementação. A tabela 16 apresenta os demais movimentos executados por cada divisão

e as condições necessárias para a execução deste movimento. A tabela 16 também está ordenada pelo primeiro estado de cada divisão (PE).

Tabela 15. Divisões do Autômato de Exploração.

<b>Estado inicial</b>	<b>Primeiro estado</b>	<b>Execução Fundamental</b>
	6	Movimento norte inicial
	7	Movimento sul em exploração crescente para leste
	8	Movimento sul complementar de leste
	9	Movimento norte em exploração crescente para leste
	10	Movimento norte complementar de leste
5	11	Movimento sul em exploração crescente para oeste
	12	Movimento sul complementar de oeste
	13	Movimento norte em exploração crescente para oeste
	14	Movimento norte complementar de oeste
	15	Movimento retorno de exploração leste
	16	Movimento retorno de exploração oeste

Tabela 16. Movimentos complementares das divisões do Autômato de Exploração.

<b>PE</b>	<b>Demais movimento (e condições necessárias)</b>
6	<b>Sul</b> (norte ocupado), <b>Leste</b> (norte e sul ocupados / leste livre), <b>Oeste</b> (norte, sul e leste ocupados / oeste livre) e <b>Parada</b> (demais casos)
7	<b>Leste</b> (sul ocupado / leste livre e inexplorado), <b>Oeste</b> (sul ocupado / leste ocupado ou explorado / oeste livre e inexplorado) e <b>Retorno</b> (demais casos)

8	<b>Norte</b> (sul ocupado), <b>Leste</b> (norte e sul ocupados / leste livre e inexplorado) e <b>Retorno</b> (demais casos)
	<b>Leste</b> (norte ocupado / leste livre e inexplorado),
9	<b>Oeste</b> (norte ocupado / leste ocupado ou explorado / oeste livre e inexplorado) e <b>Retorno</b> (demais casos)
	<b>Sul</b> (norte ocupado), <b>Leste</b> (norte e sul ocupados / leste livre e inexplorado)
10	e <b>Retorno</b> (demais casos)
	<b>Oeste</b> (norte ocupado / oeste livre e inexplorado),
11	<b>Leste</b> (norte ocupado, oeste ocupado ou explorado / leste livre e inexplorado) e <b>Retorno</b> (demais casos)
	<b>Sul</b> (norte ocupado), <b>Oeste</b> (norte e sul ocupados / oeste livre e inexplorado)
12	E <b>Retorno</b> (demais casos)
	<b>Oeste</b> (sul ocupado / oeste livre e inexplorado),
13	<b>Leste</b> (sul ocupado / oeste ocupado ou explorado / leste livre e inexplorado) e <b>Retorno</b> (demais casos)
	<b>Norte</b> (sul ocupado), <b>Oeste</b> (norte e sul ocupados / oeste livre e inexplorado)
14	e <b>Retorno</b> (demais casos)
	<b>Leste</b> (leste livre e inexplorado) e
15	<b>Oeste</b> (leste ocupado ou explorado / oeste livre e inexplorado)
	<b>Oeste</b> (oeste livre e inexplorado) e
16	<b>Leste</b> (oeste ocupado ou explorado / leste livre e inexplorado)

Na configuração inicial do Autômato de Exploração o estado inicial está conectado com a divisão PE 6 responsável pelo movimento inicial para o norte (conforme comportamento exibido pelo robô na figura 23 do exemplo do item 6.2).

Nas tabelas 15 e 16 nota-se uma primeira simetria entre as divisões PE 7 e 9 (que executam o movimento crescente para leste) e entre as divisões PE 8 e 10 (que executam o movimento complementar de deslocamento leste). Uma segunda simetria pode ser notada entre as divisões PE 7, 8, 9 e 10 (que executam os movimentos leste) e as divisões PE 11, 12, 13 e 14 (que executam os movimentos oeste). Por fim, nota-se uma terceira simetria entre as divisões de retorno PE 15 e 16.

Para que o autômato registre o caminho de exploração de um ambiente no qual o robô está inserido de forma a conseguir retornar por ele a procura de outros locais a serem mapeados – demarcando o ambiente ou não – é construída uma trilha de retorno, isto é, os passos dados pelo robô durante a exploração do local são gravados em forma de memória de pilha: o último dado gravado é o primeiro a ser recuperado. Esta pilha foi implementada no Autômato de Exploração entre os estados 3 e 4. A figura 39 exemplifica a trilha de retorno inicialmente vazia e seus sucessivos incrementos durante dois passos de exploração norte e um leste. Destaca-se que a codificação da trilha de retorno segue o exposto na tabela 14 e que o caractere registrado corresponde ao passo necessário para que o autômato retorne a posição anterior. Assim como no capítulo anterior, o símbolo  $\epsilon$  representa o caractere vazio.

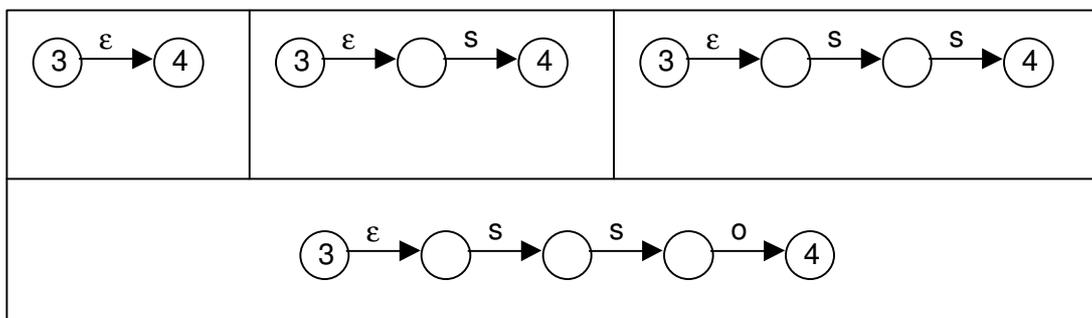


Figura 39. Trilha de retorno inicialmente vazia e seu incremento durante três passos.

As posições que são determinadas como marcos do ambiente explorado são assinaladas na trilha de retorno através de uma segunda pilha: a pilha de marcos. Esta pilha de marcos é inicialmente construída entre os estados 22 e 21 e sofre ampliação a cada marco definido através da adição de um novo estado e uma transição consumindo o caractere 'a'. Uma transição consumindo o símbolo vazio 'ε' é inserida de modo a relacionar a pilha de marcos com a pilha de retorno no estado recém criado. A figura 40 ilustra a pilha de marcos inicialmente vazia e, em seguida, com um marco assinalado no estado central da configuração final do exemplo da figura 39.

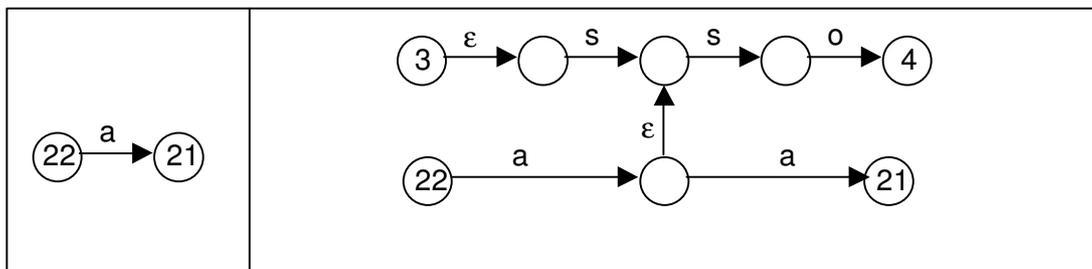


Figura 40. Trilha de marcos inicialmente vazia e seu incremento após uma demarcação.

A seguir descreve-se, de forma detalhada, como as ações adaptativas permitem a execução das diversas operações que devem ser realizadas pelo Autômato de Exploração. Tais operações são:

1. Composição da informação de saída do autômato;
2. Registro de uma trilha de retorno com informação sobre o deslocamento do robô para que este possa regressar à sua posição de origem;
3. Definição das posições especiais do ambiente com marcos e registrá-las na trilha de retorno;
4. Otimização da trilha de retorno de modo que contenha apenas informação sobre os movimentos necessários para que a trajetória de regresso do robô seja minimizada;

5. Desempilhamento da trilha de retorno durante a trajetória de regresso do robô;
6. Busca por novas áreas a serem exploradas durante o movimento de retorno do robô.

Os nomes das ações adaptativas responsáveis pelo comportamento do autômato seguem ordem alfabética, salvo exceções que ocorreram devido ao processo de elaboração do trabalho.

A figura 41 mostra a representação gráfica da divisão do Autômato de Exploração responsável pelos movimentos iniciais em direção norte (PE 6). Na figura nota-se que os nove primeiros estados (e as transições que os conectam) são destinados ao processamento da informação de vizinhança previamente registrada no mapa mas, por ser a divisão que trata dos movimentos iniciais de exploração, essa informação não modifica a computação do autômato. Os estados seguintes (e as transições que os conectam) são destinados a processar as informações obtidas pelo conjunto de sensores. Ao processar o caractere 'f' da cadeia de entrada – o qual denota que o norte está livre para movimento – o autômato executa a ação adaptativa A, que está descrita na figura 42.

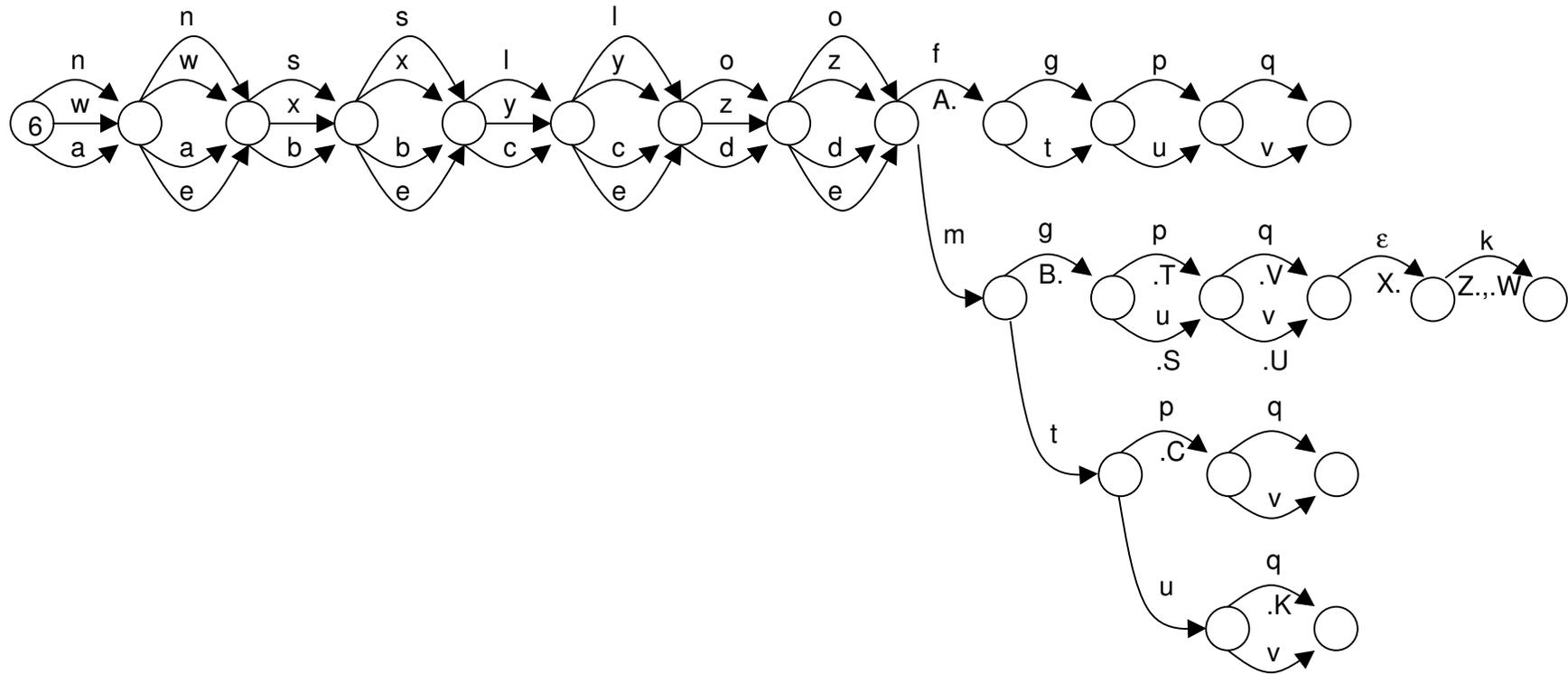


Figura 41. Representação gráfica da divisão do Autômato de Exploração responsável pelo movimento inicial para o norte.

```

A{var1,var2:
-[(1,var1)->2]
+[(1,n)->2]

-[(5,#):Y->var2]
+[(5,#):Y->6]}

```

Figura 42. Ação adaptativa A do Autômato de Exploração.

A ação A apaga qualquer transição existente entre os estados de saída (1 e 2), adiciona uma transição indicando movimentação norte e assegura que o estado inicial do autômato esteja conectado com a divisão responsável pelo movimento norte inicial (PE 6). O deslocamento inicial para o norte compõe o comportamento apresentado na figura 23 do exemplo do item 6.2. Ao encontrar um obstáculo ao norte e uma área livre ao sul, a divisão PE 6 executa a ação adaptativa B, descrita na figura 43.

```

B{var1,var2,var3,var4,ger1*:
-[(1,var1)->2]
+[(1,s)->2]

-[(5,#):Y->var2]
+[(5,#):Y->7]

-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,n)->4]}

```

Figura 43. Ação adaptativa B do Autômato de Exploração.

A ação adaptativa B indica que o próximo passo exploratório do robô será na direção sul e direciona o estado inicial (5) para a divisão do Autômato de Exploração responsável pelo movimento sul crescente para leste (PE 7). Tal processo pode ser considerado como o início da exploração do ambiente porque inaugura a construção da trilha de retorno entre os estados 3 e 4.

Existem algumas ações adaptativas similares à ação B, que estão descritas na tabela 17.

Tabela 17. Ações adaptativas semelhantes responsáveis pelo movimento exploratório crescente para leste e para oeste.

<b>Ação adaptat.</b>	<b>Função</b>
B	Deslocamento sul e estado inicial direcionado para a divisão PE 7
H	Deslocamento norte e estado inicial direcionado para a divisão PE 9
L	Deslocamento sul e estado inicial direcionado para a divisão PE 13
J	Deslocamento norte e estado inicial direcionado para a divisão PE 11

Pode-se ainda destacar no autômato apresentado na figura 41 que, se o conjunto de sensores do robô reconhecer os espaços norte e sul como ocupados e o espaço leste como livre, a ação adaptativa C será executada. Esta ação é similar à ação adaptativa B já descrita, porém indica deslocamento leste e direciona o estado inicial (5) para a divisão do Autômato de Exploração que executa o movimento sul complementar de deslocamento leste (PE 8). Esta divisão é responsável por verificar se não há nenhum espaço a ser coberto pelo robô à sul antes de retomar o deslocamento em ziguezague para norte (situação semelhante à apresentada na figura 25 do exemplo do item 6.2). Existem algumas ações adaptativas similares à ação C, que estão descritas na tabela 18.

Tabela 18. Ações adaptativas semelhantes responsáveis pelo movimento para leste e oeste.

<b>Ação adaptat.</b>	<b>Função</b>
C	Deslocamento leste e estado inicial direcionado para a divisão PE 8
I	Deslocamento oeste e estado inicial direcionado para a divisão PE 10
M	Deslocamento leste e estado inicial direcionado para a divisão PE 14
K	Deslocamento oeste e estado inicial direcionado para a divisão PE 12

A demarcação do ambiente é feita através do registro e comparação das condições ao leste e ao oeste do robô. O registro da condição da região leste é feito com a inclusão de uma transição entre os estados 17 e 18: transição consumindo o caractere '1' representa que a região está livre e transição consumindo o caractere 'o' representa que a região está ocupada. A representação da condição da região oeste é feita da mesma maneira entre os estados 19 e 20. A figura 44 ilustra uma situação em que a condição leste é livre e a condição oeste é ocupada.

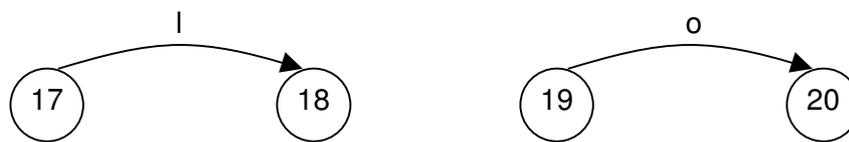


Figura 44. Representação gráfica do registro das condições de leste e oeste.

Quando uma condição de demarcação é detectada, o autômato cria mais um estado na pilha de marcos. Tal estado origina uma transição que consome o caractere  $\epsilon$  e que pode terminar em um destes dois estados: estado 23 – que indica marco de uma região com condição livre para ocupada (situação ilustrada na figura 32) – ou estado 24 – que indica marco de uma região com condição ocupada para livre (situação ilustrada na figura 31). Esta terminação é realizada de forma provisória, porque a transição deve ser concluída no estado demarcado na trilha de retorno (como exemplificado na figura 40).

A ação adaptativa T, presente no autômato apresentado na figura 41 tem a função de registrar a condição leste livre e comparar a condição atual com a condição anterior com a finalidade de definir a demarcação do ambiente. A ação adaptativa T é apresentada na figura 45 e as outras ações similares são descritas na tabela 19.

<pre>T{var1,var2: ?[(17,o)-&gt;var2] -[(17,var1)-&gt;18] +[(17,l)-&gt;18] TT(var2)}</pre>	<pre>TT(i){var1,ger1*: -[(var1,a)-&gt;21] +[(var1,a)-&gt;ger1] +[(ger1,a)-&gt;21] +[(ger1,#)-&gt;24]}</pre>
---	---

Figura 45. Ação adaptativa T do Autômato de Exploração.

A ação T inicialmente consulta a condição leste anteriormente registrada e em seguida insere uma transição que indica que a condição de leste é livre. Se a condição anterior for ocupada, a ação T chama a ação adaptativa final TT, que por sua vez incrementa a pilha de marcos com mais um estado direcionado para o estado provisório 24.

Tabela 19. Ações adaptativas semelhantes responsáveis pelo registro e comparação da condição à leste e à oeste.

<b>Ação adaptativa</b>	<b>Função</b>
T	Registro de leste livre e comparação com registro leste anterior
S	Registro de leste ocupado e comparação com registro leste anterior
V	Registro de oeste livre e comparação com registro oeste anterior
U	Registro de oeste ocupado e comparação com registro oeste anterior

A ação adaptativa X, apresentada na figura 46 tem a função de deslocar a transição do marco recém criado dos estados provisórios 23 e 24 para a posição correta na trilha de retorno, atingindo configuração final semelhante à apresentada na figura 40. A regra é necessária devido ao posicionamento correto do marco na trilha de retorno variar de acordo com a configuração detectada no ambiente (de espaço livre para espaço ocupado ou o inverso) conforme apresentado nas figuras 31 e 32.

<pre> X{var1,var2,var3,var4, var5,var6,var7: ?[(var2,var3)-&gt;4] ?[(var4,var5)-&gt;var2] -[(var1,#)-&gt;23] +[(var1,#)-&gt;var4] -[(var6,#)-&gt;24] +[(var6,#)-&gt;var2]  ?[(25,var7)-&gt;26]  XX(var1,var2,var4,var7)} </pre>	<pre> XX(i,j,k,l) {var1,var2,ger1*,ger2*:  ?[(26,var1)-&gt;25] -[(i,#)-&gt;k] -[(k,var2)-&gt;j] +[(k,var1)-&gt;ger1] +[(ger1,var2)-&gt;ger2] +[(ger2,l)-&gt;j] +[(i,#)-&gt;ger2]} </pre>
---	--

Figura 46. Ação adaptativa X do Autômato de Exploração.

A função da ação adaptativa final XX está relacionada com a modificação da trilha de retorno de acordo com a demarcação do ambiente. Portanto, tal modificação será apresentada primeiramente.

A trilha de retorno do Autômato de Exploração é construída a cada deslocamento executado pelo robô. Com a finalidade de otimizar a trilha de modo que ela contenha apenas os movimentos necessários para que o robô, ao retornar de um local já explorado, se desloque diretamente até o último marco definido, alguns registros são apagados da trilha construída. Para se apagar registros da trilha de retorno, os seguintes requisitos devem ser atendidos:

- Algum marco já deve ter sido definido na trilha;
- O último registro da trilha de retorno (identificado como n) deve ser um deslocamento para norte ou para sul;
- O registro n – 1 da trilha de retorno deve ser um deslocamento para leste ou para oeste;
- O registro n – 2 da trilha de retorno deve ser um deslocamento para norte ou para sul e não ser idêntico ao deslocamento n;
- Os registros n e n – 2 não devem estar demarcados.

Quando tais requisitos são atendidos, os registros  $n$  e  $n - 2$  da trilha de retorno são apagados com a finalidade de se evitar deslocamentos em ziguezague desnecessários entre o último marco definido e a posição atual do robô, dado que este tipo de movimentação apenas é conveniente durante o mapeamento de ambientes inexplorados. As ações adaptativas  $Z$  e  $W$ , que são executadas quando o último caractere 'k' da cadeia de entrada é processado pelo Autômato de Exploração (figura 41), têm a finalidade de verificar se todos os requisitos foram atendidos e, em caso afirmativo, apagar as duas transições. Cada ação adaptativa verifica apenas um requisito e, se este for atendido, chama uma ação adaptativa final (nomeadas  $ZZ$ ,  $ZZZ$ ,  $ZZZZ$ ,  $WW$ ,  $WWW$  e  $WWWW$ ). As figuras 47 e 48 apresentam estes conjuntos de ações adaptativas encadeadas.

$Z\{var1, var2, var3:$ $-[(25, var2) \rightarrow 26]$ $-[(26, var3) \rightarrow 25]$ $?[(var1, s) \rightarrow 4]$  $ZZ(var1)\}$	$ZZ(i)$ $\{var1, var2, var3:$ $?[(var1, var2) \rightarrow i]$ $?[(var3, n) \rightarrow var1]$  $ZZZ(i, var1, var3)\}$
$ZZZ(i, j, k)\{var1, var2, var4, var5:$ $+[(i, h) \rightarrow i]$ $+[(j, h) \rightarrow j]$ $?[(var1, a) \rightarrow 21]$ $?[(var1, \#) \rightarrow var2]$ $+[(var2, h) \rightarrow var2]$ $-[(var2, h) \rightarrow var2]$ $?[(i, var4) \rightarrow i]$ $?[(j, var5) \rightarrow j]$ $-[(i, h) \rightarrow i]$ $-[(j, h) \rightarrow j]$  $ZZZZ(i, j, k, var4, var5)\}$	$ZZZZ(i, j, k, l, m)$ $\{var1, var2, var3:$ $-[(i, s) \rightarrow 4]$ $-[(j, var1) \rightarrow i]$ $+[(j, var1) \rightarrow 4]$  $-[(k, n) \rightarrow j]$ $-[(var2, var3) \rightarrow k]$ $+[(var2, var3) \rightarrow j]$  $+[(25, s) \rightarrow 26]$ $+[(26, n) \rightarrow 25]\}$

Figura 47. Ação adaptativa  $Z$  do Autômato de Exploração.

A transição  $n$  apagada fica temporariamente gravada entre os estados 25 e 26 e a transição  $n - 2$  apagada fica temporariamente gravada entre os estados 26 e 25. Este

procedimento é necessário porque há um caso especial em que a trilha de retorno otimizada deve ser reconstruída e as duas transições apagadas podem ser recuperadas. Esta necessidade ocorre quando a trilha de retorno foi otimizada e o autômato deve assinalar um marco do tipo região anterior livre e região posterior ocupada (situação descrita na figura 32). Em tais situações, o autômato deve assinalar o marco na posição anterior da trilha de retorno e esta pode ter sido apagada. A ação adaptativa XX (figura 46) tem a função de reconstruir a trilha em tais ocorrências.

<pre> W{var1: ?[(var1,n)-&gt;4]  WW(var1) } </pre>	<pre> WW(i) {var1,var2,var3: ?[(var1,var2)-&gt;i] ?[(var3,s)-&gt;var1]  WWW(i,var1,var3)} </pre>
<pre> WWW(i,j,k){var1,var2,var4,var5: +[(i,h)-&gt;i] +[(j,h)-&gt;j] ?[(var1,a)-&gt;21] ?[(var1,#)-&gt;var2] +[(var2,h)-&gt;var2] -[(var2,h)-&gt;var2] ?[(i,var4)-&gt;i] ?[(j,var5)-&gt;j] -[(i,h)-&gt;i] -[(j,h)-&gt;j]  WWWW(i,j,k,var4,var5)} </pre>	<pre> WWWW(i,j,k,l,m) {var1,var2,var3: -[(i,n)-&gt;4] -[(j,var1)-&gt;i] +[(j,var1)-&gt;4]  -[(k,s)-&gt;j] -[(var2,var3)-&gt;k] +[(var2,var3)-&gt;j]  +[(25,n)-&gt;26] +[(26,s)-&gt;25]} </pre>

Figura 48. Ação adaptativa W do Autômato de Exploração.

A figura 49 apresenta a divisão do Autômato de Exploração PE 7, responsável pelo movimento sul em exploração crescente para o leste. As demais divisões do autômato são semelhantes a esta divisão apresentada.

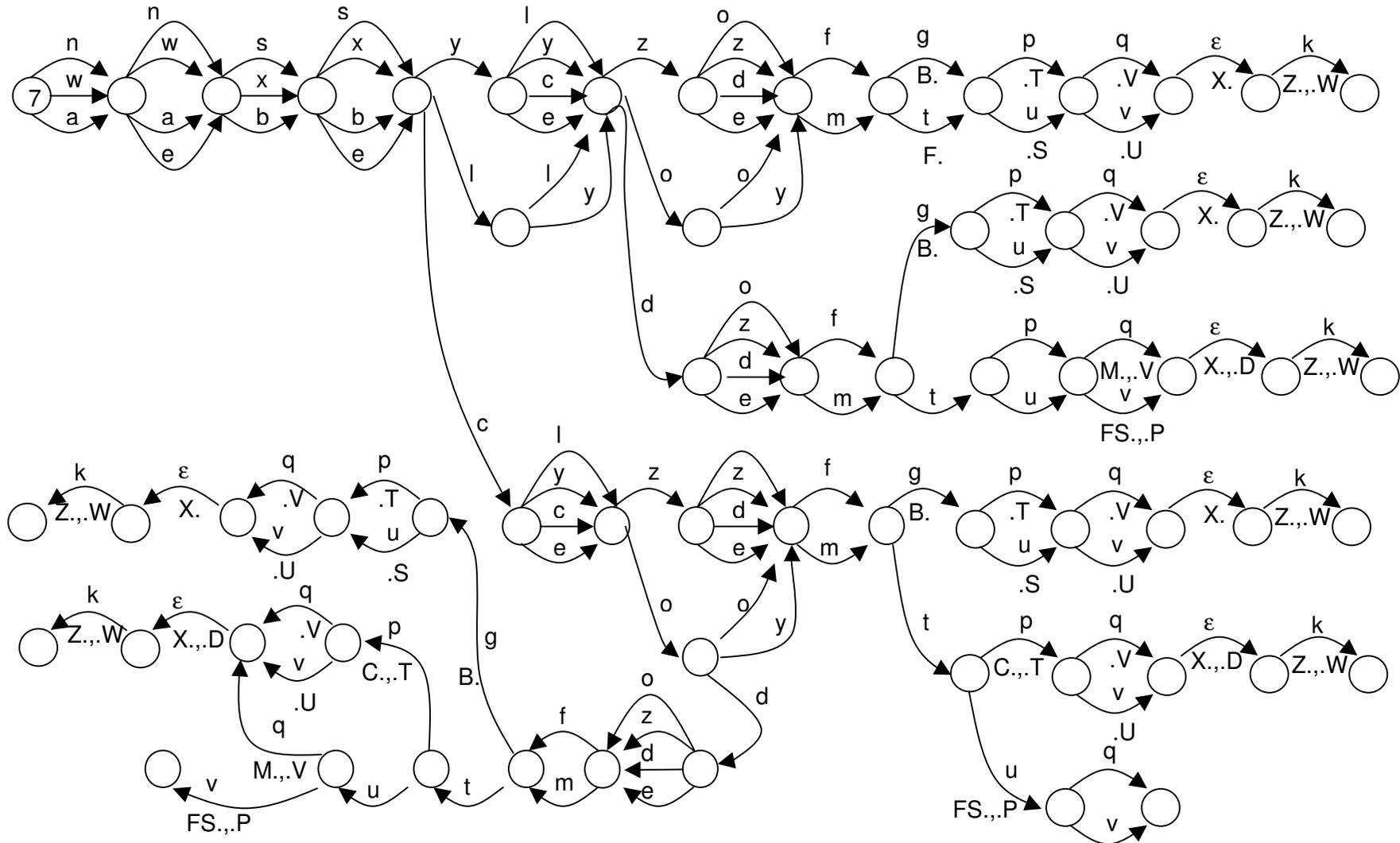


Figura 49. Representação gráfica da divisão do Autômato de Exploração responsável pelo movimento sul em exploração crescente para leste.

Diferentemente da divisão PE 6 (apresentada na figura 41), a divisão PE 7 processa a informação de vizinhança previamente registrada no mapa para poder separar quatro diferentes categorias: (i) leste e oeste já explorados; (ii) leste explorado e oeste inexplorado; (iii) leste inexplorado e oeste explorado e; (iv) leste e oeste inexplorados. O processamento tem como finalidade prover as condições necessárias para que a divisão realize as demais funções a ela atribuídas (relacionadas na tabela 16).

A divisão PE 7 possui ainda a ação adaptativa D, a qual é executada todas as vezes que o autômato determina que o próximo movimento seja na direção leste ou oeste. Esta ação tem como função apagar os registros das condições de leste e oeste (figura 44) porque a comparação destas condições não deve ser realizada após tais deslocamentos – comparações apenas são realizadas em deslocamentos nas direções norte e sul. A figura 50 apresenta a ação adaptativa D.

$  \begin{aligned}  &D\{\text{var1}, \text{var2}, \text{var3} : \\  &-\{ (17, \text{var1}) \rightarrow 18 \} \\  &-\{ (19, \text{var2}) \rightarrow 20 \}  \end{aligned}  $
---

Figura 50. Ação adaptativa D do Autômato de Exploração.

A sinalização de saída que exprime que um estado é marco é realizada através da inserção de uma transição com o caractere ‘r’ entre os estados 2 e 1. O Sistema Gerente é responsável por copiar esta informação para a cadeia de entrada do Autômato de Mapeamento com o objetivo que o estado assinalado seja demarcado no mapa. A figura 51 apresenta um exemplo no qual a saída do Autômato de Exploração indica deslocamento sul e que o estado deve ser demarcado.

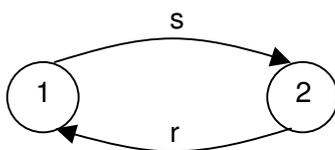


Figura 52. Representação gráfica da informação de saída demarcada do Autômato de Exploração.

A divisão PE 7 possui ainda a ação adaptativa F que está apresentada na figura 51.

<pre>F{var1,var2,var3, var4,var5,var6,var7,va r8,var9,var10: -[(1,var1)-&gt;2] ?[(var2,var3)-&gt;4] ?[(var4,a)-&gt;21] ?[(var4,var5)-&gt;var2] -[(var6,var7)-&gt;4] -[(var8,var9)-&gt;var6] +[(var8,var9)-&gt;4] +[(1,var7)-&gt;2] -[(5,#):Y-&gt;var10] +[(5,#):Y-&gt;15] FF(var5)}</pre>	<pre>FF(e){var1,var2,var3,var4,var5: +[(2,r)-&gt;1] -[(var1,a)-&gt;21] -[(var1,#)-&gt;var2] -[(var3,a)-&gt;var1] +[(var3,a)-&gt;21] ?[(var4,a)-&gt;21] ?[(var4,var5)-&gt;4] FFF(var5)} FFF(e){var1,var2,var3,var4,var5: -[(var1,a)-&gt;21] -[(var1,#)-&gt;4] -[(var2,a)-&gt;var1] +[(var2,a)-&gt;21] ?[(var4,a)-&gt;21] ?[(var4,var5)-&gt;4] FFF(var5)}</pre>
---	---

Figura 51. Ação adaptativa F do Autômato de Exploração.

A ação adaptativa F é uma ação de retorno, isto é, desempilha a última transição da trilha de retorno, copia esta informação para os estados de saída do Autômato de Exploração e direciona o estado inicial para a divisão de retorno de leste PE 15 (conforme tabela 15). Além destas funções, a ação adaptativa F também verifica se a transição da trilha de retorno é um estado demarcado, através da verificação da pilha de marcos. Em caso afirmativo ela chama a ação adaptativa final FF.

A ação adaptativa FF tem a função de inserir a transição 'r' entre os estados 1 e 2 e apagar o último registro da pilha de marcos. Se, devido à configuração do ambiente, mais de um marco for atribuído ao mesmo estado, a ação adaptativa FFF é chamada.

A ação adaptativa FFF tem como função apagar os registro duplicados da pilha de marcos.

Na divisão PE 7, apresentada na figura 49, verifica-se uma variante da ação adaptativa de retorno F: a ação adaptativa FS. Esta ação adaptativa e suas ações adaptativas finais (FFS e FSS) são descritas na figura 53 e têm a função de executar o deslocamento de retorno em uma situação do ambiente considerada como exceção: a trilha de retorno foi otimizada de forma que o último deslocamento da pilha indica oeste e o autômato deve assinalar um marco do tipo região leste anteriormente livre e posteriormente ocupada. Neste caso particular, a transição oeste não deve ser desempilhada da trilha de retorno, o autômato deve indicar deslocamento norte para que o robô continue a exploração do ambiente crescente para leste.

<pre> FS{var1,var2,var3,var4, var5,var6,var7,var8: ?[(17,l)-&gt;var2] -[(17,var1)-&gt;18] +[(17,o)-&gt;18]  -[(var3,var4)-&gt;4] -[(var5,var6)-&gt;var3] +[(var5,var6)-&gt;4] +[(1,var4)-&gt;2]  -[(5,#):Y-&gt;var7] +[(5,#):Y-&gt;15]  FFS(var2)} </pre>	<pre> FSS(i){var1,var2,var4, var5,var6,ger1*,ger2*: -[(var1,#)-&gt;24] -[(var1,a)-&gt;21] -[(var4,a)-&gt;var1] +[(var4,a)-&gt;21]  -[(1,var2)-&gt;2] +[(1,n)-&gt;2] +[(2,r)-&gt;1]  -[(var5,var6)-&gt;4] +[(var5,var6)-&gt;ger1] +[(ger1,s)-&gt;ger2] +[(ger2,o)-&gt;4]} </pre>
<pre> FSS(i){var1,var2,ger1*: -[(var1,a)-&gt;21] +[(var1,a)-&gt;ger1] +[(ger1,a)-&gt;21] +[(ger1,#)-&gt;24] ?[(1,o)-&gt;var2]  FSS(var2)} </pre>	

Figura 53. Ação adaptativa FS do Autômato de Exploração.

As ações de retorno semelhantes à F e à FS estão descritas na tabela 20.

Tabela 20. Ações adaptativas semelhantes responsáveis pelo deslocamento de retorno.

<b>Ação adaptat.</b>	<b>Função</b>
F	Desempilha trilha de retorno e direciona estado inicial para PE 15
G	Desempilha trilha de retorno e direciona estado inicial para PE 16
FS	Verifica se a situação de retorno é exceção leste livre-ocupado quando deslocamento sul e direciona estado inicial para PE 15
FN	Verifica se a situação de retorno é exceção leste livre-ocupado quando deslocamento norte e direciona estado inicial para PE 15
GS	Verifica se a situação de retorno é exceção leste livre-ocupado quando deslocamento sul e direciona estado inicial para PE 16
GN	Verifica se a situação de retorno é exceção leste livre-ocupado quando deslocamento norte e direciona estado inicial para PE 16

As ações de retorno em situações de exceção (quatro últimas ações relacionadas na tabela 20) são complementadas pela ação adaptativa P, que possui a função de inserir a transição com o caractere 'r' entre os estados 2 e 1 se o estado de retorno for um marco. A figura 54 apresenta a ação adaptativa P (e sua ação adaptativa final PP).

<pre>P{var1, var2: ?[(var1, a)-&gt;21] ?[(var1, var2)-&gt;4] PP(var2)}</pre>	<pre>PP(i){var1, var2, var3: +[(2, r)-&gt;1] -[(var1, a)-&gt;21] -[(var1, #)-&gt;4] -[(var2, a)-&gt;var1] +[(var2, a)-&gt;21] ?[(var2, var3)-&gt;4] PP(var3)}</pre>
--	---

Figura 54. Ação adaptativa P do Autômato de Exploração.

A transição que interliga o estado original com os primeiros estados das divisões do Autômato de Exploração consome o caractere vazio ‘ $\epsilon$ ’ e possui a ação adaptativa Y. Esta ação adaptativa possui a função de garantir que a sinalização de marco da operação anterior do autômato não permaneça em sua nova utilização. A figura 55 apresenta a ação adaptativa Y.

$Y\{ : \\ -[(2, r) \rightarrow 1] \}$
---------------------------------------

Figura 55. Ação adaptativa Y do Autômato de Exploração.

Finalmente, o último grupo de ações adaptativas existentes no Autômato de Exploração estão representadas pela ação adaptativa N na figura 56. A ação N é utilizada pela divisão PE 8, quando após um deslocamento leste há uma região sul complementar a ser coberta antes do prosseguimento da exploração em ziguezague para o norte. Esta ação indica que o próximo movimento deverá ser na direção sul e garante que o estado inicial do autômato conduza à divisão PE 8.

$N\{var1, var2, var3, var4, ger1* : \\ -[(1, var1) \rightarrow 2] \\ +[(1, s) \rightarrow 2] \\ \\ -[(5, \#) : Y \rightarrow var2] \\ +[(5, \#) : Y \rightarrow 8] \\ \\ -[(var3, var4) \rightarrow 4] \\ +[(var3, var4) \rightarrow ger1] \\ +[(ger1, n) \rightarrow 4] \}$
--

Figura 56. Ação adaptativa N do Autômato de Exploração.

A tabela 21 relaciona as ações adaptativas semelhantes à ação adaptativa N.

Tabela 21. Ações adaptativas semelhantes responsáveis pelo movimento complementar após deslocamento para leste e para oeste.

<b>Ação adaptat.</b>	<b>Função</b>
N	Deslocamento sul e estado inicial direcionado para a divisão PE 8
O	Deslocamento norte e estado inicial direcionado para a divisão PE 10
R	Deslocamento sul e estado inicial direcionado para a divisão PE 14
Q	Deslocamento norte e estado inicial direcionado para a divisão PE 12

#### **6.4 Considerações sobre o Autômato de Exploração**

De acordo com a configuração do Autômato de Exploração apresentada, nota-se que o autômato pode exibir dois comportamentos distintos na busca espaços ainda não explorados: (i) retornar de locais já mapeados seguindo a mesma trajetória em ziguezague definida durante a exploração – comportamento apresentado nas figuras 27 a 30 – ou (ii) retornar por uma trajetória que interliga locais especiais, demarcados segundo a configuração do ambiente explorado – comportamento apresentados nas figuras de 34 a 37. Para exibir o primeiro comportamento o autômato utiliza um grupo de ações adaptativas e para exibir o segundo comportamento utiliza esse mesmo grupo adicionado de um outro grupo de ações adaptativas distintas. A tabela 22 apresenta a quantidade total de ações adaptativas envolvidas em cada um dos dois comportamentos.

Ainda, de acordo com a figura 36, pode-se notar que o comportamento de retorno de locais já mapeados segundo a trajetória definida entre marcos não é um comportamento exibido em

procedimentos de navegação de robôs, nos quais há um ponto de início e um ponto de destino e o robô deve utilizar a melhor trajetória possível de deslocamento. O deslocamento apresentado pelo robô é característico da busca por áreas ainda não mapeadas, o que configura tarefa essencial para a aplicação de mapeamento.

Tabela 22. Quantidade de ações adaptativas envolvidas nos comportamentos de retorno.

<b>Comportamento</b>	<b>Quantidade de ações adaptativas</b>
Trajectoria de retorno em ziguezague	15
Trajectoria de retorno utilizando marcos	53

Deve-se destacar que o algoritmo proposto para a exploração de ambientes, descrito neste capítulo, permite que o robô alcance todas as regiões livres (isto é, todos os espaços que não estejam obstruídos e possam ser conectados com a posição inicial de exploração do robô por uma trajetória exequível) em qualquer ambiente no qual esteja inserido devido à capacidade de exibição do comportamento descrito a seguir:

1. Se o ambiente a ser explorado for um retângulo de quaisquer dimensões, livre de quaisquer obstáculos e o robô estiver inicialmente posicionado no canto superior esquerdo deste ambiente, o algoritmo permitirá ao robô desenvolver uma trajetória em ziguezague que cubra toda a superfície do ambiente e em seguida retorne à sua posição inicial.
2. Se o robô não estiver inicialmente posicionado na parte superior do retângulo, o algoritmo permitirá que o robô primeiramente se desloque até tal posição para, em seguida, iniciar a exploração do local.

3. Se o robô não estiver inicialmente posicionado na lateral esquerda do ambiente, o algoritmo permitirá que o robô explore a porção do retângulo localizada a leste e, após seu retorno à posição inicial, explore a porção do retângulo localizada a oeste.
4. Se o ambiente no qual o robô estiver inserido não for um retângulo ou não estiver livre de obstáculos, o algoritmo demarcará todas as posições que exibirem características irregulares quando o robô passar por elas e, após o término da exploração da porção do espaço em que este se encontra, conduzirá o robô de volta a todas as posições demarcadas. Em cada uma destas posições o algoritmo verificará se o local ainda não foi explorado e se o caminho para fazê-lo se encontra livre. Neste caso, conduzirá o robô a explorá-lo seguindo os mesmos princípios já expostos. Caso contrário, ou quando o robô terminar a exploração desta porção do espaço, o algoritmo conduzirá o robô à próxima posição demarcada no ambiente até que o robô se encontre novamente em sua posição inicial de exploração.

## 7. SIMULAÇÕES

Objetivando a validação, teste e análise dos resultados obtidos com a proposta das estruturas de mapeamento e exploração fundamentadas nos Autômatos Adaptativos e do modelo de transmissão de informação apresentado no capítulo 3, construiu-se um simulador, composto de por dois blocos distintos:

1. O primeiro bloco é responsável pela simulação de Autômatos Adaptativos<sup>3</sup> e é compatível com a teoria apresentada no trabalho de Neto (1993).
2. O segundo bloco é responsável pela ativação alternada dos autômatos de mapeamento e exploração a cada movimento realizado pelo robô. Este bloco também permite a construção de ambientes a serem explorados, a determinação do posicionamento inicial do robô móvel e a verificação dos deslocamentos deste. Tal ambiente é construído em uma matriz binária, na qual '0' significa espaço livre e '1' significa espaço obstruído.

A seguir são descritas algumas das simulações realizadas. A escolha dos ambientes simulados a serem utilizados neste trabalho foi realizada baseada na representatividade de algumas propriedades específicas em cada uma das simulações, de modo que tais propriedades possam abranger grande parte das possíveis situações.

### 7.1 Primeira simulação: ambiente com características simples

O primeiro ambiente simulado possui características que podem ser consideradas como triviais:

---

<sup>3</sup> Este programa está baseado em uma primeira versão construído pelo aluno Eduardo Costa.

espaço quadrado com apenas um obstáculo, também quadrado, no centro. A figura 57a apresenta o ambiente da primeira simulação e a posição inicial do robô, indicada pelo círculo listrado.

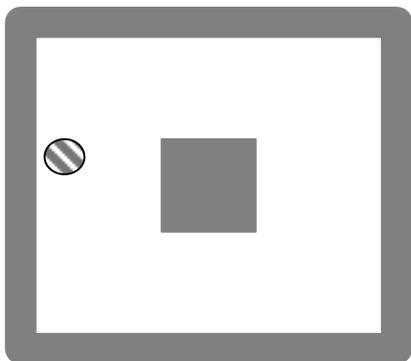


Figura 57a. Ambiente da primeira simulação.

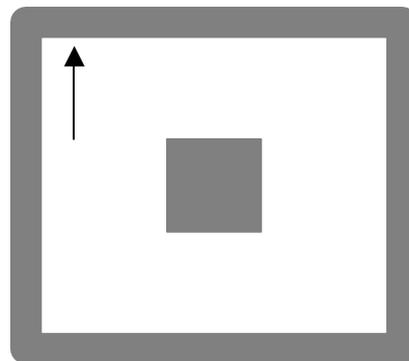


Figura 57b. Deslocamento inicial do robô.

A figura 57b representa o deslocamento norte primário do robô antes do início dos movimentos exploratórios. A figura 58a apresenta a exaustão da trajetória de exploração na porção do espaço em que o robô está inserido e a figura 58b apresenta o percurso de retorno do robô (indicada pela linha tracejada) até a porção do espaço ainda não explorado. Destaca-se que todas as simulações apresentadas empregam a estratégia de uso de marcos no ambiente.

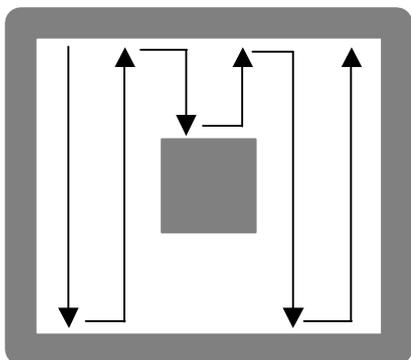


Figura 58a. Exploração inicial do ambiente da primeira simulação.

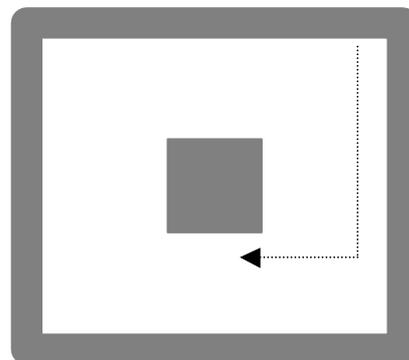


Figura 58b. Deslocamento de retorno.

A figura 59a apresenta a exploração do trecho final do ambiente e a figura 59b, o deslocamento de retorno com a finalidade de procurar por espaços ainda não explorados, segundo a trajetória entre marcos.

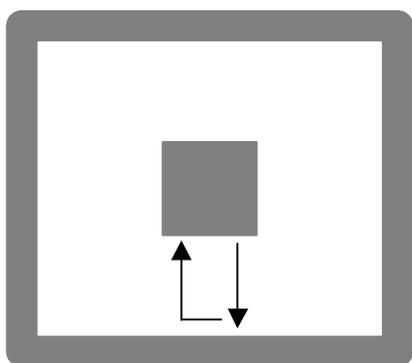


Figura 59a. Exploração final do ambiente da primeira simulação.

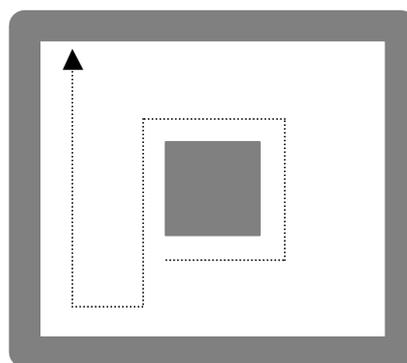


Figura 59b. Deslocamento de retorno até a posição de início de exploração.

A figura 60a apresenta a sobreposição das trajetórias de exploração descritas nas figuras 58a e 59a com a finalidade de confirmar a total exploração do ambiente e a figura 60b apresenta a posição dos marcos definidos para este ambiente.

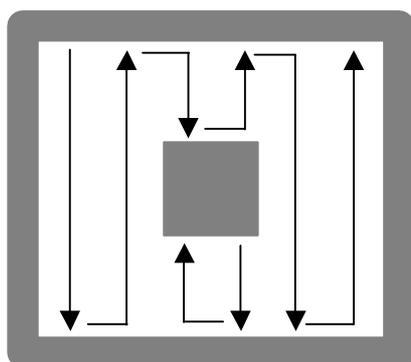


Figura 60a. Sobreposição das trajetórias de exploração da primeira simulação.

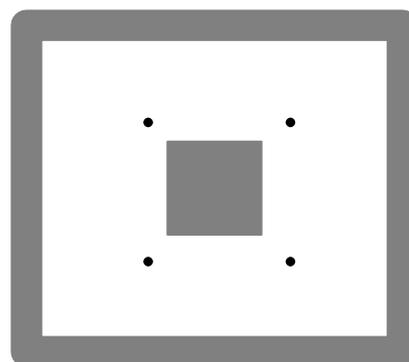


Figura 60b. Demarcação do ambiente da primeira simulação.

A figura 61 mostra a configuração final do Autômato de Mapeamento após a completa exploração do ambiente da primeira simulação. As setas duplas representam transições indicativas de locais livres para deslocamento, as linhas espessas representam transições indicativas de locais obstruídos e as linhas simples representam transições indicativas de locais não mapeados. O círculo escuro assinala a posição final de exploração.

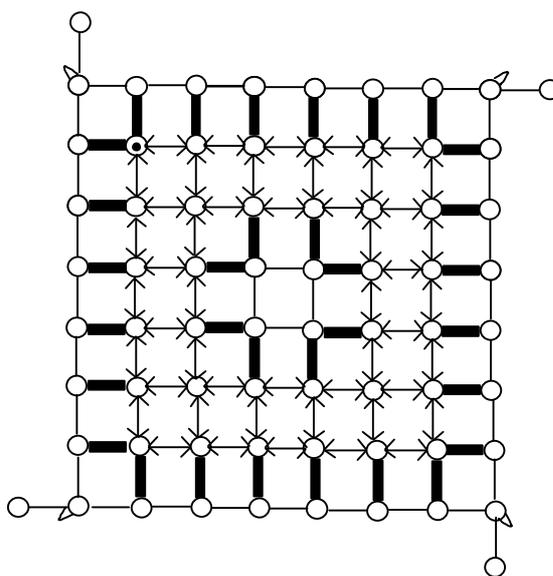


Figura 61. Representação da configuração final do Autômato de Mapeamento após a conclusão da exploração do ambiente da primeira simulação.

## 7.2 Segunda simulação: ambiente típico

O segundo ambiente simulado possui características que podem ser consideradas como típicas para espaços de ambientes internos, como um escritório ou uma sala: ambiente retangular, com um obstáculo irregular no centro e irregularidades nas paredes. A figura 62a representa o ambiente da segunda simulação e a posição do início da exploração do robô. A figura 62b apresenta o movimento de exploração complementar ao deslocamento leste.

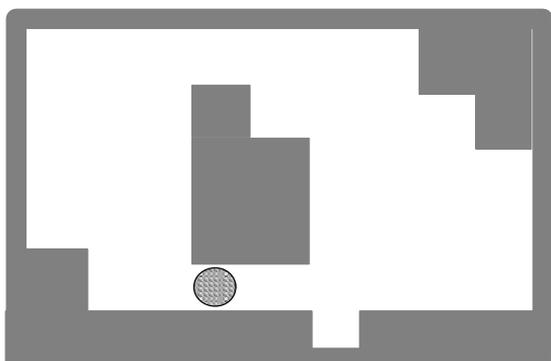


Figura 62a. Ambiente da segunda simulação.

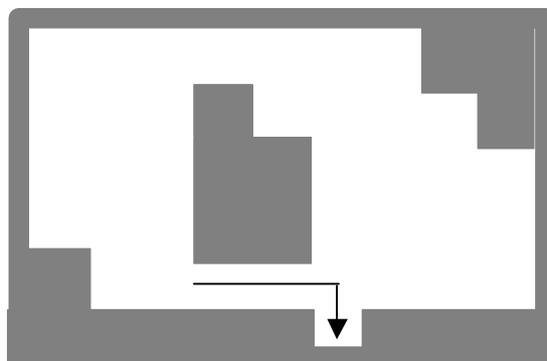


Figura 62b. Exemplo de deslocamento complementar.

A figura 63a apresenta o movimento de exploração crescente para leste e a figura 63b apresenta o movimento de retorno em direção a uma região ainda não explorada. Estas duas figuras representam uma situação exceção, conforme descrito no capítulo 6.

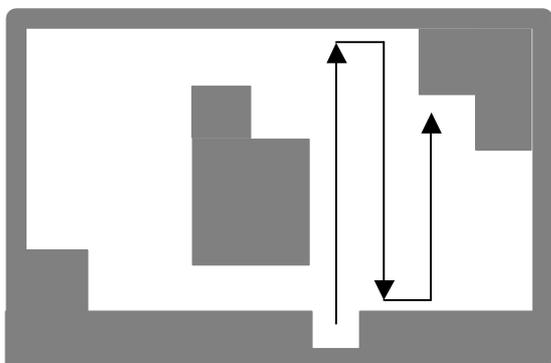


Figura 63a. Exploração crescente para leste na segunda simulação.



Figura 63b. Retorno em direção a uma região ainda não explorada.

A figura 64a apresenta o deslocamento de exploração da região alcançada no percurso representado na figura 63b. A figura 64b apresenta o deslocamento do robô entre as posições já definidas como marcos com o objetivo de procurar por regiões ainda não exploradas.



Figura 64a. Exaustão da trajetória leste de exploração.

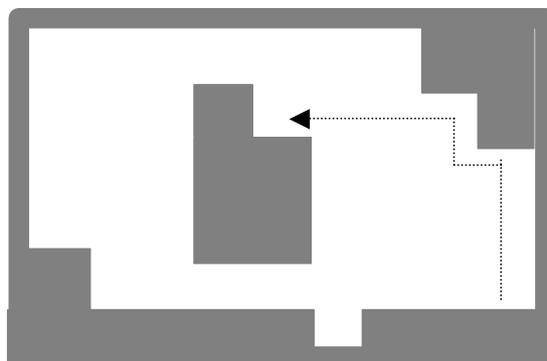


Figura 64b. Percurso de retorno entre marcos.

A figura 65a apresenta os deslocamentos de exploração na direção oeste e a figura 65b apresenta a trajetória de retorno realizada pelo robô até sua posição inicial.

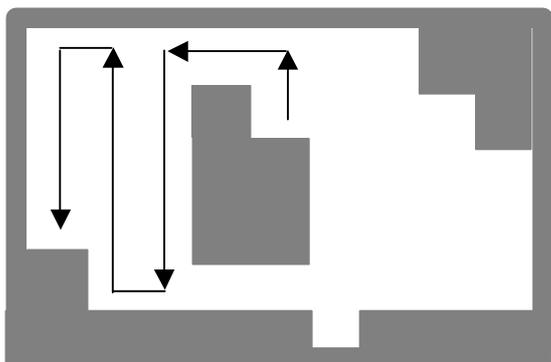


Figura 65a. Exaustão da trajetória oeste de exploração.

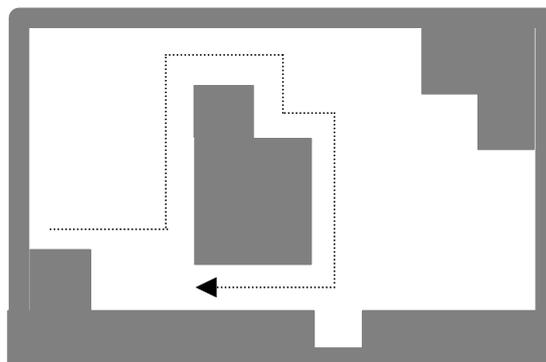


Figura 65b. Percurso de retorno até a posição inicial de exploração.

A figura 66a representa a sobreposição das trajetórias de exploração descritas nas figuras 62b, 63a, 64a e 65a com a finalidade de apresentar a total exploração do ambiente e a figura 66b apresenta a posição dos marcos definidos para este ambiente. Na figura destaca-se a não existência do marco que seria posicionado no canto inferior esquerdo do obstáculo no centro do ambiente, devido à detecção de ambiente já explorado durante a passagem do robô pelo local.

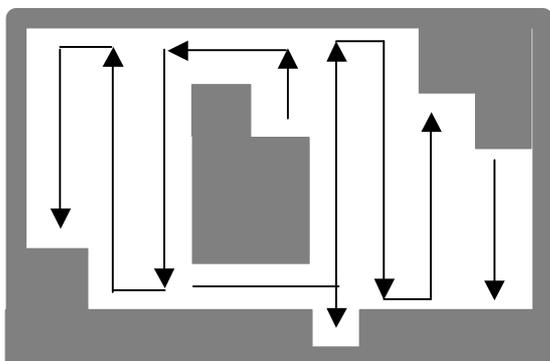


Figura 66a. Sobreposição das trajetórias de exploração da segunda simulação.

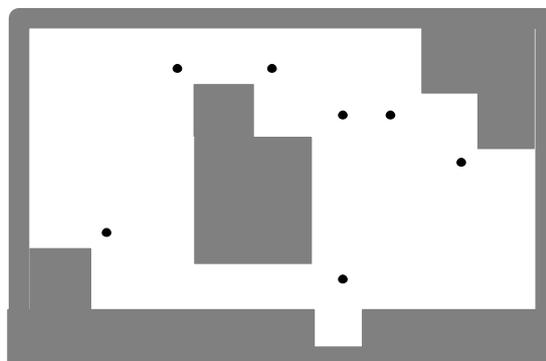


Figura 66b. Demarcação do ambiente da segunda simulação.

A figura 67 mostra a configuração final do Autômato de Mapeamento após a exploração completa do ambiente da segunda simulação. A representação das transições desta figura (e de todas figuras similares) segue o padrão utilizado na figura 61.

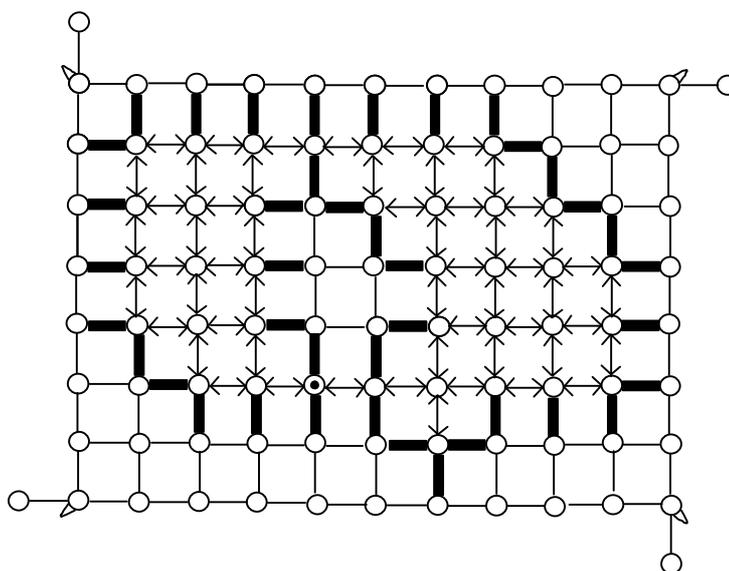


Figura 67. Representação da configuração final do Autômato de Mapeamento após a conclusão da exploração do ambiente da segunda simulação.

### 7.3 Terceira simulação: ambiente atípico

O terceiro ambiente simulado possui características distintas dos ambientes comuns: é constituído

por diversos corredores interligados, de largura igual às dimensões do robô e de formatos irregulares. A figura 68a representa o ambiente da terceira simulação e a posição inicial do robô. A figura 68b representa o deslocamento inicial de exploração deste ambiente.

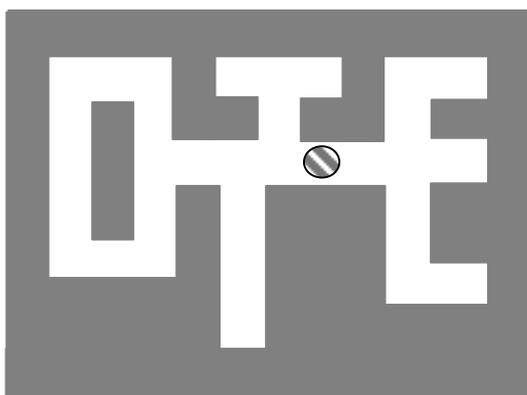


Figura 68a. Ambiente da terceira simulação.



Figura 68b. Deslocamento inicial de exploração.

As figuras 69a, 69b e 70a apresentam os sucessivos movimentos de exploração e retorno da porção do ambiente localizado a leste da posição inicial de exploração. As figuras 70b e 71a apresentam os deslocamentos necessários para a exploração da região localizada ao norte da posição inicial de exploração.

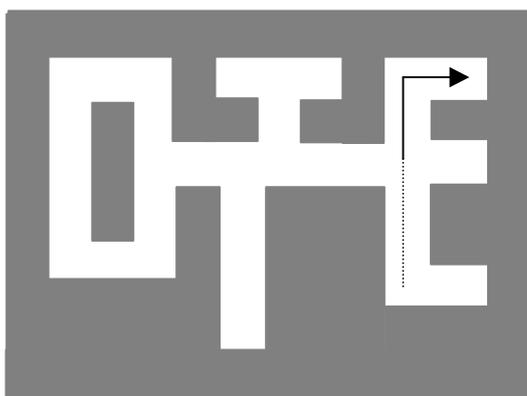


Figura 69a. Exploração da região superior direita da terceira simulação.

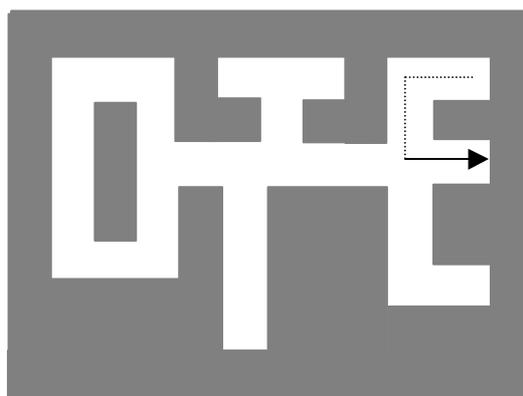


Figura 69b. Exploração da região central direita da terceira simulação.

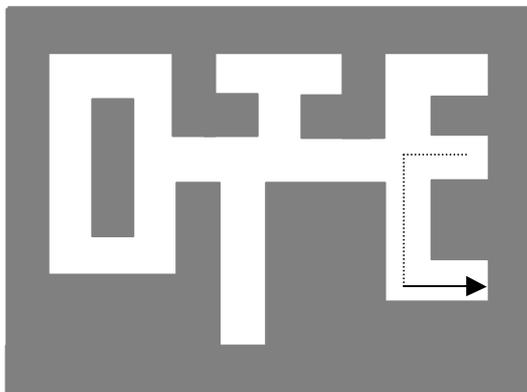


Figura 70a. Exploração da região inferior direita da terceira simulação.

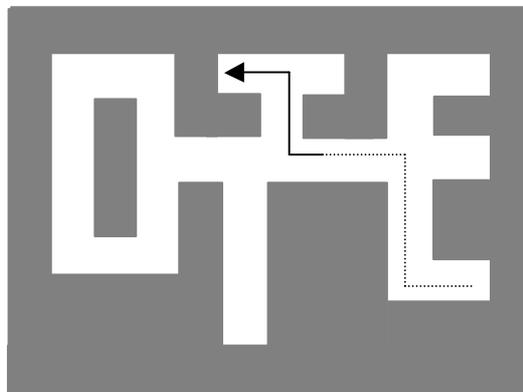


Figura 70b. Exploração inicial da região superior da terceira simulação.

A figura 71b apresenta a exploração da região do ambiente localiza ao sul da posição inicial de exploração.

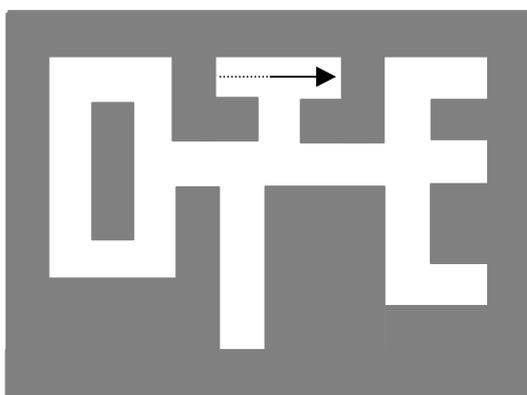


Figura 71a. Exploração final da região superior da terceira simulação.

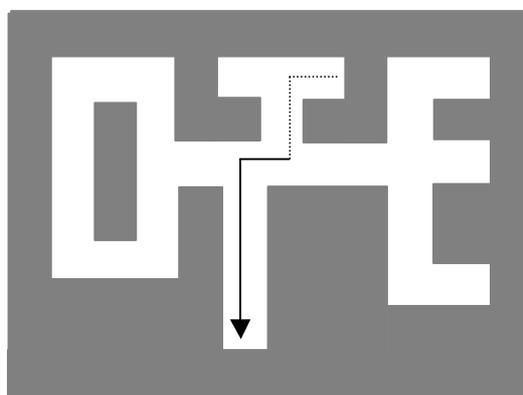


Figura 71b. Exploração da região inferior da terceira simulação.

As figuras 72a e 72b apresentam os sucessivos movimentos de exploração e retorno da porção do ambiente localizado ao oeste da posição inicial de exploração.

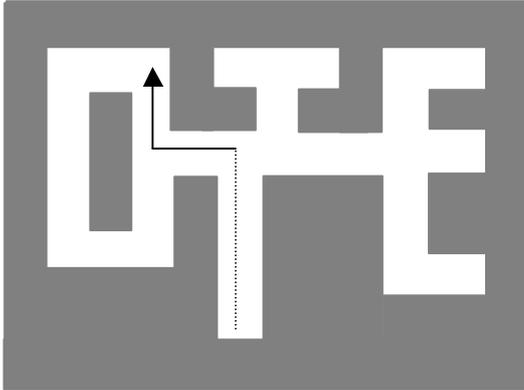


Figura 72a. Exploração inicial da região esquerda da terceira simulação.

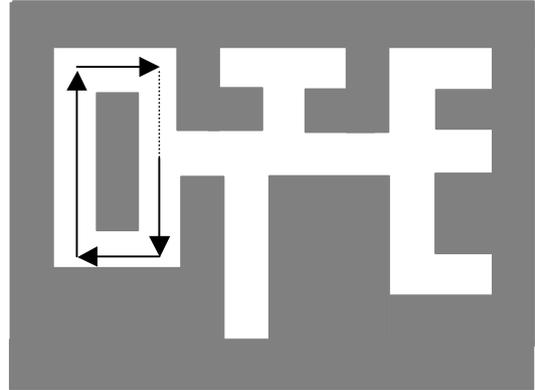


Figura 72b. Exploração final da região esquerda da terceira simulação.

A figura 73a apresenta o percurso de retorno para a posição inicial de exploração e a figura 73b representa a sobreposição das trajetórias de exploração descritas nas figuras 68b a 73b com a finalidade de apresentar a total exploração do ambiente. A figura 74 representa as posições sinalizadas com marcos e a figura 75 mostra a configuração final deste autômato após a completa exploração do ambiente da terceira simulação.

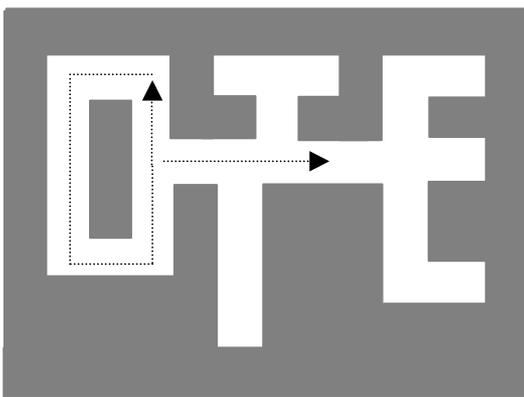


Figura 73a. Trajetória de retorno da última região explorada no ambiente da terceira simulação.

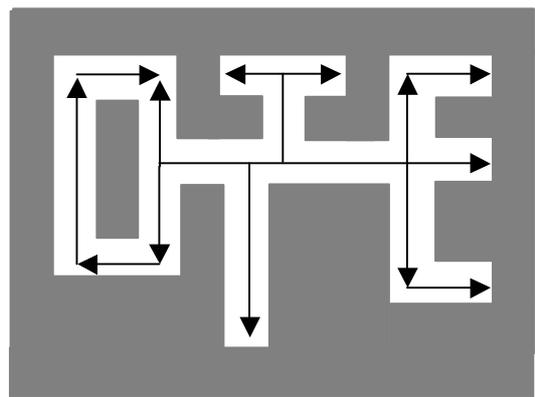


Figura 73b. Sobreposição das trajetórias de exploração da terceira simulação.



deslocamento inicial no sentido contrário a tal posição. Impedimentos que apresentam tais configurações são também conhecidos como obstáculos em ‘C’ (HWANG; JU, 2002).

A figura 76a representa o ambiente da quarta simulação e a posição de início do robô. A figura 76b representa a exploração da primeira porção do espaço deste ambiente e a figura 76c, o percurso de retorno desta região.

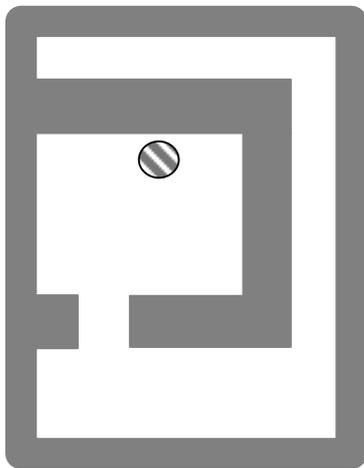


Figura 76a. Ambiente da quarta simulação.

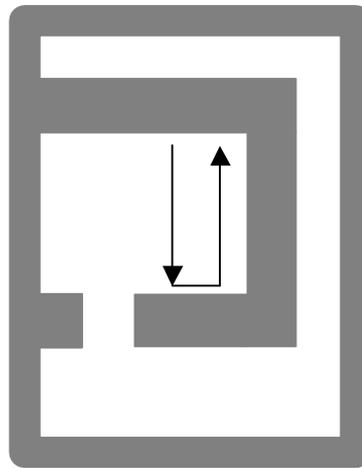


Figura 76b. Exploração da primeira porção do espaço da quarta simulação.

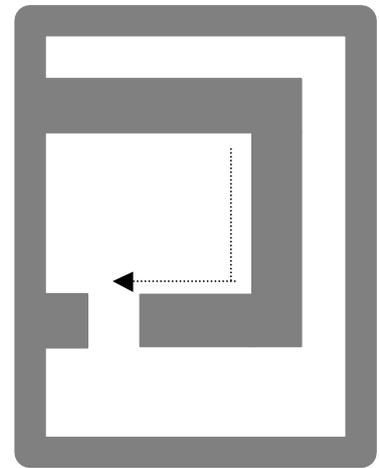


Figura 76c. Trajetória de retorno da primeira porção do espaço da quarta simulação.

A figura 77a apresenta a trajetória de exploração complementar de movimento oeste e as figuras 77b e 77c, a exploração da região exterior do ambiente.

A figura 78a apresenta a trajetória de retorno entre as posições demarcadas para a exploração da última porção do espaço da quarta simulação (figura 78b). A figura 78c apresenta o percurso de retorno final até a posição de início de exploração do robô.

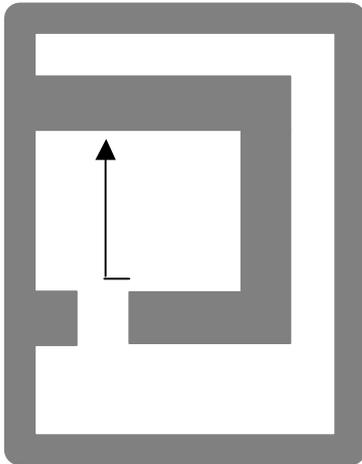


Figura 77a. Deslocamento complementar de movimento oeste.

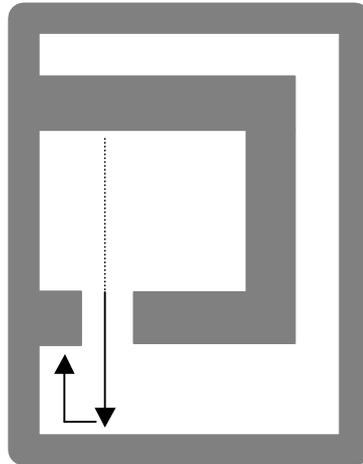


Figura 77b. Exploração inicial da região exterior da quarta simulação.

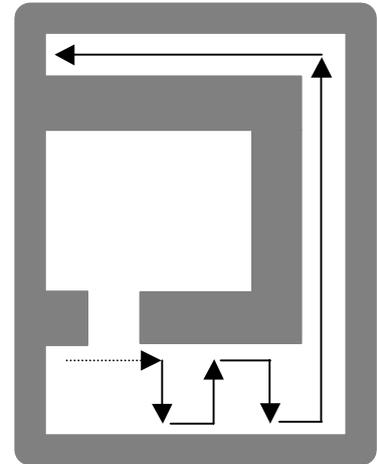


Figura 77c. Exploração final da região exterior da quarta simulação.

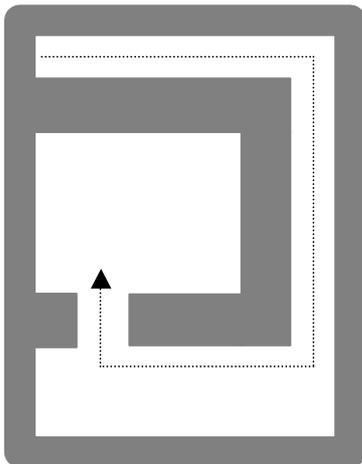


Figura 78a. Deslocamento de retorno.

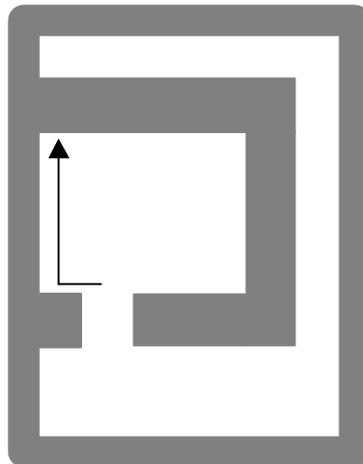


Figura 78b. Exploração da porção final do ambiente da quarta simulação.

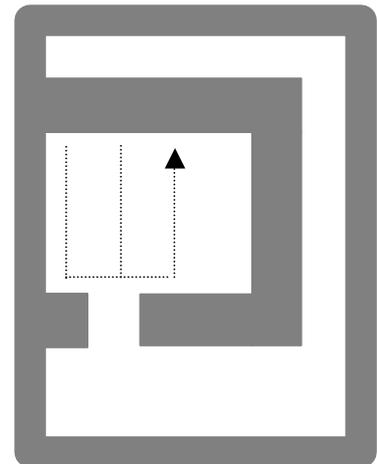


Figura 78c. Deslocamento de retorno em direção à posição inicial de exploração.

As figuras 79a e 79b apresentam, respectivamente, a sobreposição de todas as trajetórias de exploração do ambiente da quarta simulação (figuras 76b, 77a, 77b, 77c e 78b) e os marcos definidos para este ambiente.

A figura 80 mostra a configuração final do Autômato de Mapeamento após a exploração completa do ambiente da quarta simulação.

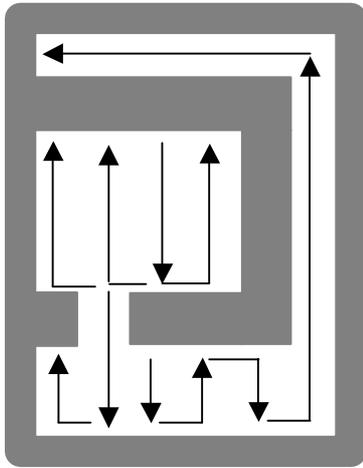


Figura 79a. Sobreposição das trajetórias de exploração da quarta simulação.

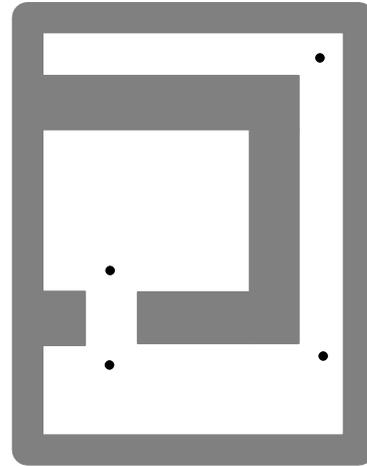


Figura 79b. Demarcação do ambiente da quarta simulação.

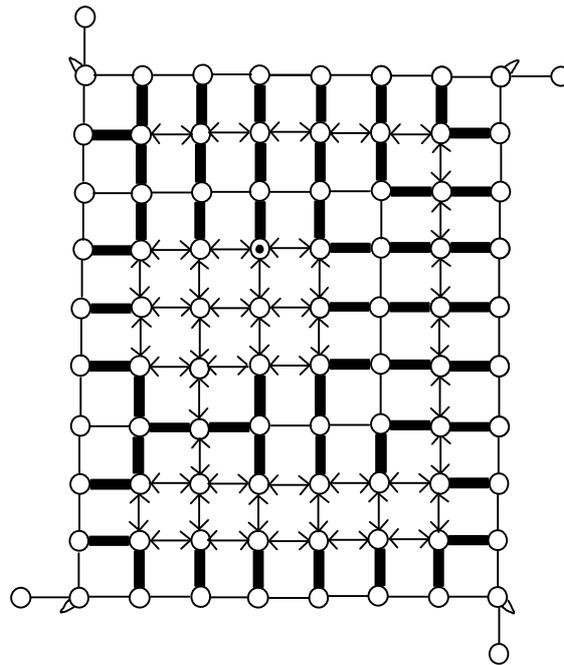


Figura 80. Representação da configuração final do Autômato de Mapeamento após a conclusão da exploração do ambiente da quarta simulação.

### 7.5 Quinta simulação: ambiente-teste de deslocamento contínuo

Como a estratégia proposta para a cobertura total de ambientes emprega um deslocamento em

zigzague, o quinto ambiente simulado foi escolhido para testar o comportamento do robô em situações nas quais este tipo de deslocamento fosse impossibilitado. A figura 81a representa o ambiente da última simulação e o robô inicia a exploração na posição central da espiral. A figura 81b representa a exaustão da trajetória de exploração deste ambiente.

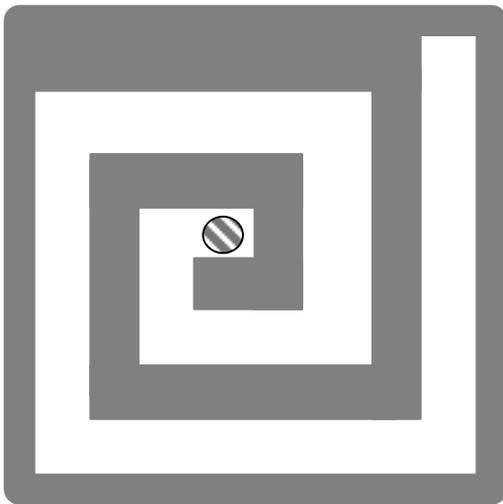


Figura 81a. Ambiente da quinta simulação.

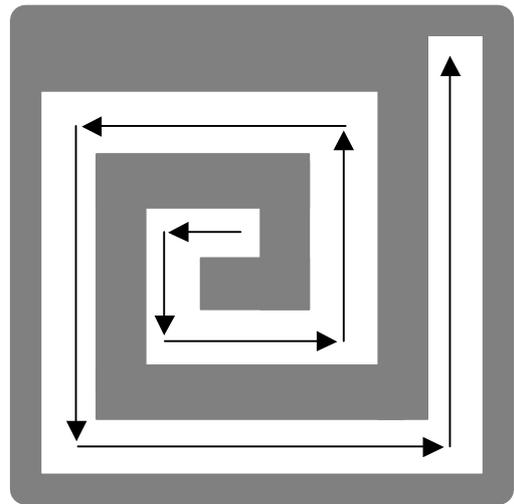


Figura 81b. Exaustão da trajetória de exploração.

As figuras 82a e 82b apresentam, respectivamente, a trajetória de retorno em direção à posição inicial de exploração e as posições definidas como marcos nesta última simulação.

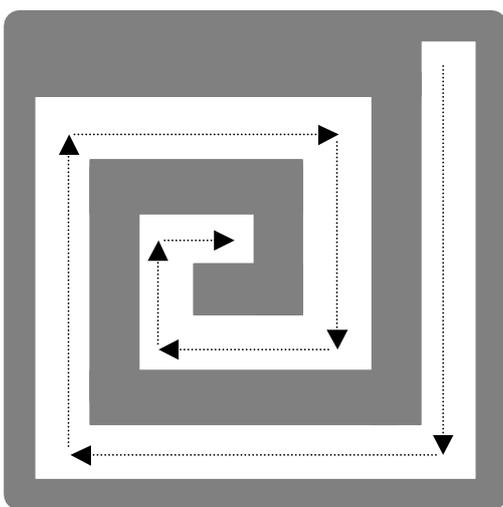


Figura 82a. Trajetória de retorno da quinta simulação.

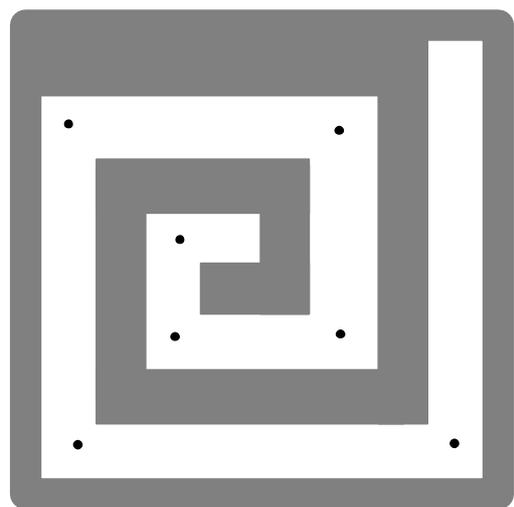


Figura 82b. Marcos definidos durante a exploração do ambiente da quinta simulação.

A figura 89 mostra a configuração final do Autômato de Mapeamento após a completa exploração do ambiente da quinta simulação.

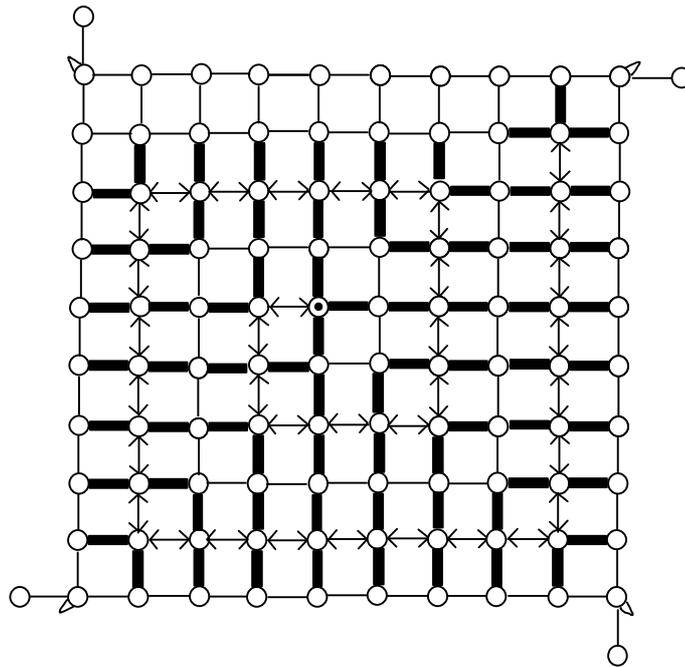


Figura 89. Representação da configuração final do Autômato de Mapeamento após a conclusão da exploração do ambiente da quinta simulação.

## 7.6 Considerações

As simulações apresentadas foram executadas em um processador Pentium III 750 MHz e, durante estes experimentos, alguns dados foram coletados para que uma análise do comportamento do modelo proposto pudesse ser realizada. A primeira informação coletada é a quantidade de passos necessários para a cobertura do ambiente, isto é, a quantidade de deslocamentos realizados pelo robô para explorar toda região livre e alcançável no espaço em que ele estava inserido. Esta informação está contida na tabela 23.

Tabela 23 – Quantidade de deslocamentos realizados pelo robô para explorar inteiramente o ambiente.

Simulação	Passos	
	Estratégia de utilização de marcos empregada	Estratégia de utilização de marcos não empregada
1ª Simulação	54	62
2ª Simulação	58	72
3ª Simulação	76	76
4ª Simulação	69	86
5ª Simulação	70	70

De acordo com o esperado, nota-se que na 1ª, 2ª e 4ª simulações, apresentadas na tabela 23, o Algoritmo de Exploração indica uma menor quantidade de deslocamentos para cobrir inteiramente os ambientes quando se emprega a estratégia de demarcação do ambiente em relação a quando não se utiliza tal estratégia. Também se observa que na 3ª e 5ª simulações o número de passos dados, empregando-se ou não a demarcação do ambiente, foi idêntico. Este fato se deve à configuração singular destes ambientes que não permite trajetórias diferenciadas, ou seja, independente da estratégia adotada nestes dois ambientes, o percurso de exploração é único e delimita a quantidade de passos necessários para explorá-lo.

Outra informação obtida nas simulações é o tempo consumido por cada uma das duas estratégias de exploração para completar o mapeamento do ambiente. Esta informação está apresentada na tabela 24.

Tabela 24 – Tempo consumido na exploração completa do ambiente.

Simulação	Tempo (segundos)	
	Estratégia de utilização de marcos empregada	Estratégia de utilização de marcos não empregada
1° Simulação	721	740
2° Simulação	830	906
3° Simulação	1054	957
4° Simulação	907	1052
5° Simulação	985	901

A tabela 24 permite observar que, também de acordo com o esperado, na 1ª, 2ª e 4ª simulações o tempo consumido para a completa exploração do ambiente é menor quando se emprega a estratégia de marcos em relação a quando não se utiliza tal estratégia. Entretanto, na 3ª e 5ª simulações ocorre a situação inversa. Tal fato se deve à característica limitante da quantidade de deslocamentos destes ambientes – que iguala as trajetórias de retorno de locais já explorados – e à maior quantidade de ações adaptativas que a estratégia de demarcação de ambientes emprega (segundo apresentado na tabela 22 do capítulo 6) – que provoca um maior consumo de tempo na exploração de tais ambientes. Esta diferença de tempo consumido no deslocamento, devido à quantidade de ações adaptativas envolvidas na execução de cada uma das estratégias, é melhor destacada na tabela 25.

Tabela 25 – Média de tempo consumido por deslocamento.

Simulação	Média de tempo (segundos) / passo	
	Estratégia de utilização de marcos empregada	Estratégia de utilização de marcos não empregada
1° Simulação	13,35	11,93
2° Simulação	14,31	12,58
3° Simulação	13,86	12,26
4° Simulação	13,14	12,23
5° Simulação	14,07	12,87

Nota-se na tabela 25 que o tempo consumido em cada ciclo de mapeamento e definição de deslocamento é maior quando se emprega a estratégia de marcos em relação a quando não se utiliza tal estratégia.

Com a finalidade de permitir uma melhor observação dos dados colhidos durante as simulações, gráficos foram construídos com estas informações. O gráfico apresentado na figura 90 relaciona a quantidade de passos necessários para a completa exploração dos ambientes com o tempo consumido para explorá-los, empregando-se a estratégia de marcos. A figura 91 mostra o gráfico desta mesma relação, porém sem o emprego da estratégia de demarcação de ambientes nas simulações. A figura 92 apresenta as duas relações comparadas no mesmo gráfico e a figura 93 apresenta o gráfico da média de tempo consumido por deslocamento realizado.

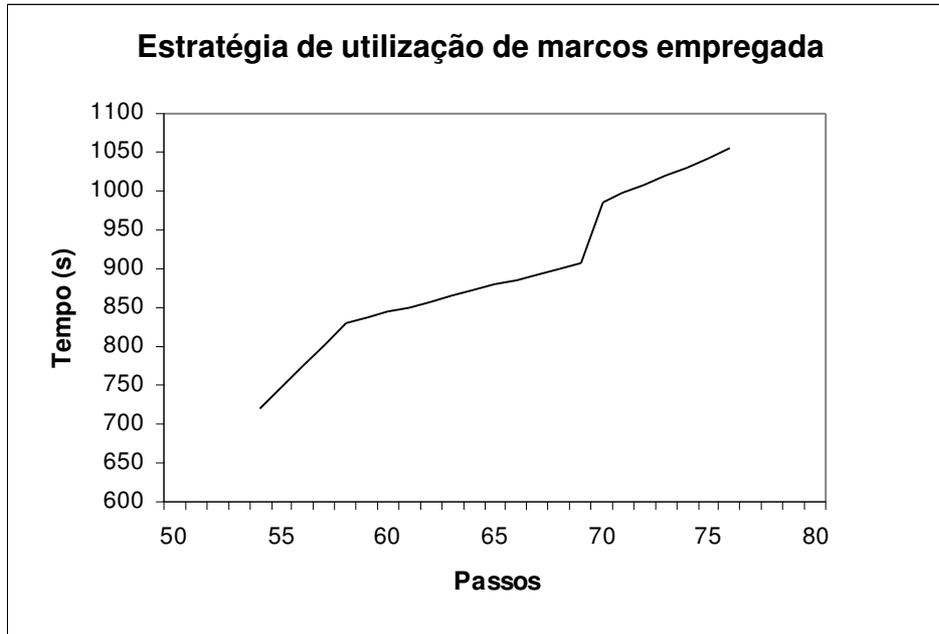


Figura 90. Gráfico do tempo consumido em função do deslocamento empregando-se a demarcação do ambiente.

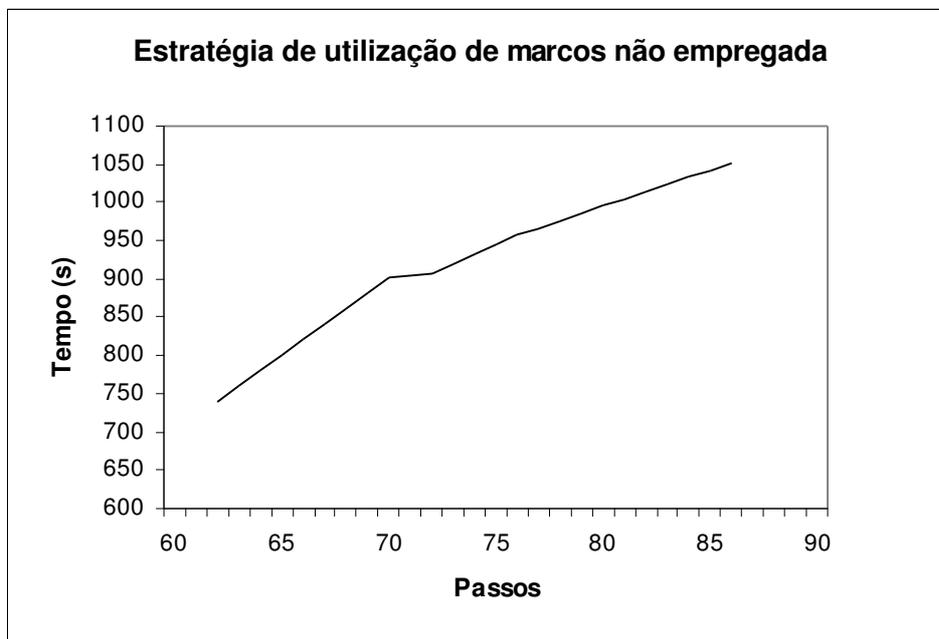


Figura 91. Gráfico do tempo consumido em função do deslocamento não se empregando a demarcação do ambiente.

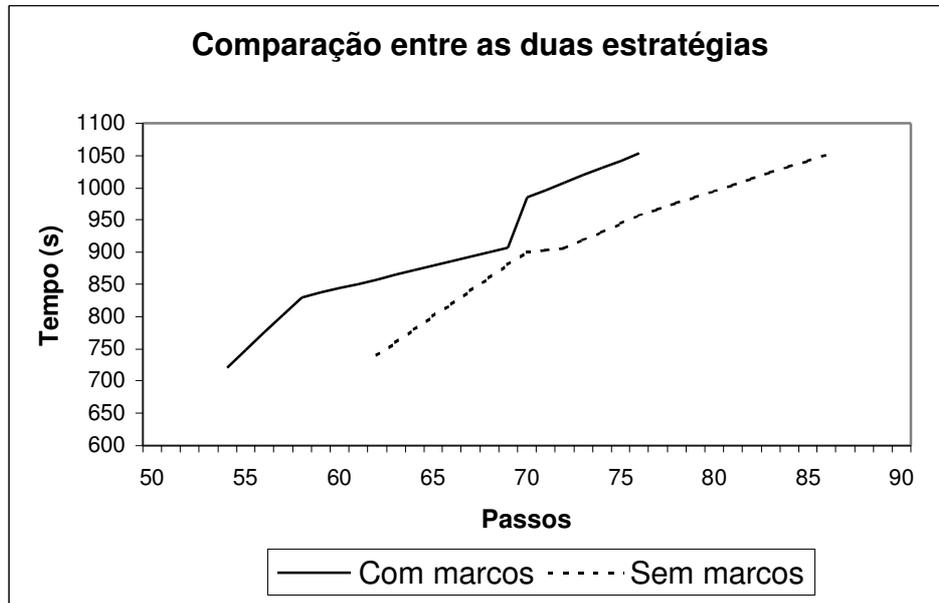


Figura 92. Comparação do tempo consumido em função do deslocamento entre as diferentes estratégias.

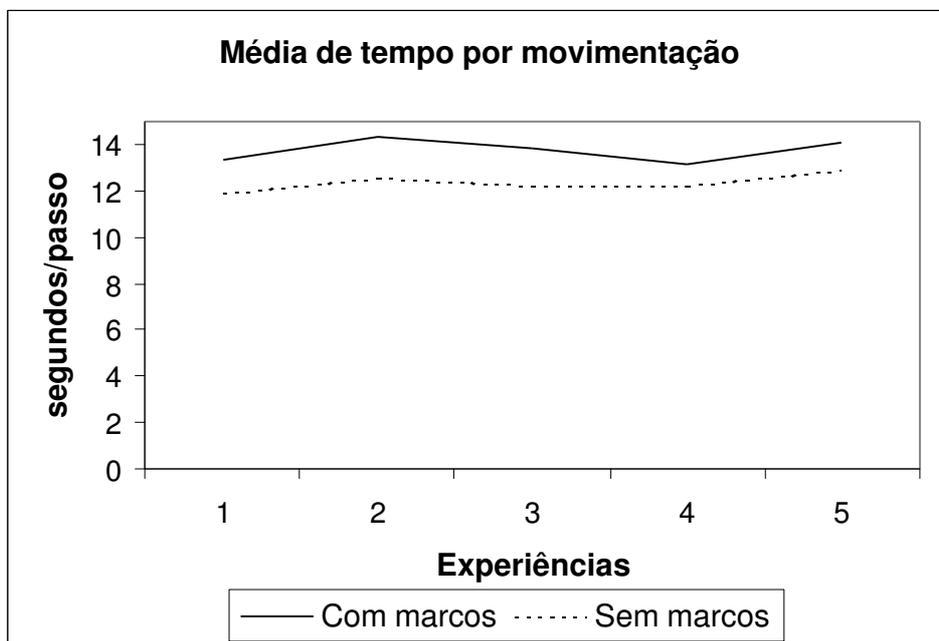


Figura 93. Comparação do tempo consumido por deslocamento entre as diferentes estratégias.

Deve-se destacar na figura 92 que, em ambas estratégias, o tempo consumido na exploração dos ambientes cresce de forma aproximadamente linear com o aumento da quantidade de

deslocamentos necessários para o mapeamento completo destes ambientes. Nota-se ainda, na figura 93, que se considerando todas as simulações efetuadas, o tempo consumido por deslocamento realizado não sofre intensas alterações em nenhuma das duas estratégias.

A tabela 26 apresenta a comparação entre as duas estratégias de definição do percurso de retorno empregadas nas simulações considerando-se apenas os ambientes nos quais há uma diferença de desempenho no deslocamento do robô (1ª, 2ª e 4ª simulações).

Tabela 26 – Comparação de médias da 1ª, 2ª e 4ª simulações entre as duas estratégias empregadas.

Atributo	Estratégia de utilização de marcos empregada	Estratégia de utilização de marcos não empregada
Média de tempo / deslocamento	13,6 segundos / passo	12,25 segundos / passo
Média de movimentação	60,33 passos	73,33 passos
Média de tempo consumido	819,33 segundos	899,33 segundos

Observando-se a tabela 26 pode-se concluir que, em ambientes nos quais há diferença de percurso, a estratégia de demarcação consome mais tempo para realizar cada deslocamento do que a estratégia de não demarcação, entretanto conclui o processo de exploração completa do ambiente com um menor número de deslocamentos, o que acarreta em uma exploração do ambiente mais veloz.

Contudo, deve-se destacar que, em aplicações reais de robótica móvel, a aplicação que emprega a demarcação do ambiente apresenta características ainda mais vantajosas, pois permite executar o

mapeamento do ambiente com menor movimentação do robô, o que diminui problemas causados por derrapagens.

O problema de derrapagem ocorre quando a informação de movimentação, definida pelo sistema de processamento do robô e transmitida para o seu sistema de deslocamento, não é efetuada com exatidão devido às irregularidades no terreno em que o robô está posicionado. O acúmulo do erro entre a posição real do robô no ambiente e a posição representada em seu mapa acarreta a necessidade da execução de outra tarefa característica da robótica móvel, a localização. A localização pode ser definida como o processo realizado pelo robô para conhecer seu próprio posicionamento em um dado ambiente através de, por exemplo, medições efetuadas por este robô no espaço ao redor (LIM; LEONARD, 2000).

Observa-se ainda nas figuras que contêm a configuração final do Autômato de Mapeamento após a completa exploração dos ambientes propostos em cada simulação (figuras 61, 67, 75, 80 e 89) que a configuração é composta por transições que indicam locais obstruídos, transições que indicam espaços livres e transições sem nenhuma informação incluída, que denotam áreas não exploradas. Estas últimas transições, chamadas de transições vagas, são necessárias para se completar o formato retangular exibido pelo Autômato de Mapeamento. Com a finalidade de se conhecer a eficiência da representação abstrata deste autômato, construiu-se um gráfico que indica a relação desta característica com um ambiente quadrado padrão sem nenhum obstáculo. A figura 94 mostra a variação da eficiência de representação 'N', obtida pela divisão entre a quantidade de transições que contêm informação pela quantidade total de transições, em função da quantidade de estados 'Q' utilizados na representação o ambiente. A variável 'N' está apresentada em porcentagem.

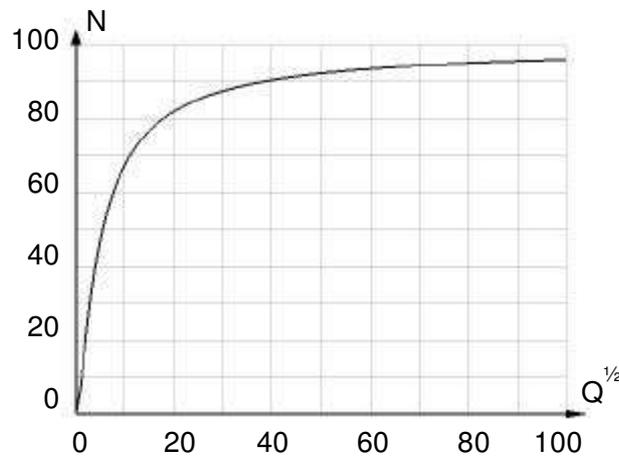


Figura 94. Gráfico da variação da eficiência de representação do Autômato de Mapeamento em função da quantidade de estados utilizados na representação do ambiente.

Nota-se que em ambientes cuja representação utiliza quantidades maiores que 120 estados (isto é, com lado do quadrado de representação igual a 11 estados), obtém-se uma eficiência de representação de mais de 70% e que esta eficiência tende a estabilizar em aproximadamente 96%.

Por fim, destaca-se que a memória ocupada pelo sistema para o registro das informações coletadas no ambiente e sua representação na forma de mapa de tal ambiente é proporcional à quantidade de transições que indicam áreas livres, áreas ocupadas e áreas não mapeadas, isto significa que a memória utilizada no processo de mapeamento é proporcional à área mapeada e às transições vagas necessárias para completar o formato retangular do Autômato de Mapeamento. Além disso, devido à característica adaptativa inerente ao formalismo escolhido para a construção da estrutura de representação de dados, a memória utilizada para o registro do mapa tem uma ocupação inicial mínima, proporcional aos nove estados e às transições que os interligam (segundo a configuração inicial do Autômato de Mapeamento apresentada na figura 4) e é progressivamente expandida de acordo com a aquisição de informações do ambiente durante o processo de exploração e mapeamento realizado pelo robô.

## 8. CONCLUSÕES

Este trabalho apresentou uma alternativa aos métodos clássicos utilizados no mapeamento robótico e na exploração de ambientes desconhecidos, através da proposta de uma abordagem para estas questões de elevada importância na área de pesquisa da robótica móvel fundamentada no emprego dos Autômatos Adaptativos.

A arquitetura apresentada pode ser classificada como uma estratégia híbrida de representação de ambientes, pois o mapa construído registra explicitamente dados sobre as conexões entre espaços distintos do espaço e sua aplicação na definição de trajetórias de deslocamento utiliza posições especiais demarcadas no ambiente (estratégias classificadas como topológicas) e, no entanto, conserva informações sobre a disposição e as dimensões dos objetos que compõem o ambiente mapeado (estratégias classificadas como geométricas).

O emprego da arquitetura proposta apresenta, como vantagens, a possibilidade da construção da representação abstrata de ambientes – os quais podem exibir topologias simples ou complexas – sem nenhum conhecimento prévio de informações sobre o local, permitindo que o robô móvel percorra todos os espaços livres e alcançáveis e adote qualquer posição como o ponto inicial de sua trajetória de exploração. O formalismo adaptativo adotado permite ainda que a área de memória ocupada pelo sistema seja utilizada de acordo com a aquisição de informações sobre a configuração do local durante sua exploração, e que o tempo consumido na exploração dos ambientes apresente um crescimento de forma aproximadamente linear de acordo com o aumento da quantidade de deslocamentos necessários para o mapeamento destes ambientes.

Por fim, pode-se concluir que este trabalho demonstrou a possibilidade do emprego dos Autômatos Adaptativos como uma ferramenta eficaz e confiável na representação abstrata de ambientes físicos e que sua utilização, a partir do desenvolvimento desta pesquisa, abre inúmeras

possibilidades de continuidade e evolução da área de pesquisa de mapeamento de ambientes desconhecidos por robôs móveis.

## **9. TRABALHOS FUTUROS**

O presente trabalho consiste na primeira pesquisa realizada com o objetivo de implementar as estruturas de mapeamento e exploração empregando-se os Autômatos Adaptativos. Portanto, diversos complementos e continuações podem ser originadas a partir desta proposta inicial. O item a seguir descreve a mais evidente delas, que é o processo de navegação utilizando o mapa construído e o item posterior relaciona outras possibilidades que se apresentaram durante a pesquisa realizada para o desenvolvimento deste trabalho.

### **9.1 Navegação**

Embora o escopo do presente trabalho não inclua o processo de navegação, realizou-se um estudo de métodos e estratégias utilizadas no planejamento de trajetória de robôs móveis que empregam um mapa do ambiente previamente construído, porque o objetivo da construção de modelos do ambiente por agentes robóticos – como o apresentado no presente trabalho – é permitir melhores deslocamentos e execuções de tarefas em tais ambientes através da aplicação de tais modelos. A aplicação do mapa construído no processo de navegação foi apresentada no trabalho Sousa e Hirakawa (2005) e está descrita resumidamente a seguir.

A estratégia de navegação proposta realiza dois tipos de processamento do mapa, um “off-line” e outro “on-line”, tarefa que pode ser considerada típica no procedimento de planejamento de trajetórias. O processamento “off-line” consiste em calcular as trajetórias que interligam os marcos assinalados pelo Autômato de Exploração e registrados no mapa pelo Autômato de Mapeamento durante a exploração do ambiente. Este cálculo entre dois estados considerados marcos pode ser realizado por um sistema gerador de seqüências de deslocamentos, que origine

distintas cadeias de entrada para serem processadas por um autômato cujos estados inicial e final sejam dois estados distintos definidos como marcos no Autômato de Mapeamento. Utilizando-se o autômato que registra o mapa para aceitar ou rejeitar as seqüências de deslocamento, descartam-se as cadeias recusadas e mantêm-se as cadeias aceitas que denotam os percursos entre dois marcos. O processamento “on-line” é executado de forma semelhante, com o objetivo de se determinarem as cadeias que representam os movimentos necessários entre a posição inicial de deslocamento e o estado demarcado mais próximo a este ponto e entre a posição final de deslocamento e o estado demarcado mais próximo a este ponto. A composição destas duas cadeias com os percursos de interligação de marcos previamente calculados representa a trajetória de deslocamento a ser executada pelo robô. Desta forma, os marcos definidos para um certo ambiente, que executam a função de apoio no processo de definição da trajetória de retorno durante a fase de exploração e mapeamento, exercem a função de submetas (em inglês sub-goals) durante o processo de navegação, isto é, posições intermediárias entre os pontos de início e fim da trajetória de movimentação do robô (HWANG et al., 2003).

O registro dos estados demarcados no Autômato de Mapeamento apresentado neste trabalho é realizado relacionando-os com um único estado especial. O processo de planejamento de trajetória pode ser aperfeiçoado pelo agrupamento, em forma de pilha, dos estados demarcados, com a finalidade de manter um registro da ordem de definição destes marcos durante o movimento exploratório e, assim, minimizar seu processamento de definição de rotas entre os marcos.

## 9.2 Outras perspectivas

A lista descrita a seguir contém a relação das perspectivas de continuidade da pesquisa desenvolvida no presente trabalho.

- A representação do ambiente realizada pelo Autômato de Mapeamento mantém uma relação entre cada estado criado e um ponto no ambiente. Pode-se utilizar o comportamento adaptativo do formalismo escolhido para executar o agrupamento de vários estados em apenas um que contenha a informação de sua representação, definindo-se, por exemplo, o perímetro de um retângulo cujo interior contenha apenas áreas livres para a movimentação e mantendo-se a relação com restante do mapa através das transições conectadas a estados vizinhos. Pode-se também representar uma área maior do mapa que contenha um obstáculo em seu interior, através da preservação do registro da área ocupada que deve ser subtraída da área livre denotada pelo estado que registra as informações do perímetro do espaço. Caso o robô realize deslocamentos pela área reduzida, utiliza-se uma ação adaptativa para expandir novamente os estados agrupados. Tal estratégia de representação trará uma economia da memória utilizada pelo sistema de mapeamento.
- Pode-se implementar a estratégia utilizada no trabalho de Marco, Prattichizzo e Vicino (2000) de exclusão de áreas não importantes do mapa construído. A operação pode ser realizada com naturalidade pelos Autômatos Adaptativos, inclusive com a exclusão das transições que não contém informação após a completa exploração do ambiente.

- A pesquisa de Zimmer (2000) permite variar a quantidade de neurônios de uma rede neural artificial conforme a necessidade de detalhamento de uma porção do espaço do mapa. O mapa construído pelo Autômato de Mapeamento também pode representar o ambiente com diferentes densidades de estados, criados de acordo com a resolução necessária para o registro de informações: locais complexos ou de grande volume de movimentação requerem maior densidade de representação do que locais simples e de pouca movimentação.
- O trabalho proposto classifica as áreas do ambiente em obstruídas, livres para deslocamento ou não mapeadas. Aliado ao registro de áreas utilizando-se diferentes resoluções de representação, é possível criar uma quarta condição de classificação, a exemplo de González et al. (1996), em áreas parcialmente ocupadas, que permita ao robô realizar uma movimentação limitada em tais porções do espaço.
- Deve-se realizar também um estudo da complexidade computacional envolvida nos processos de mapeamento e exploração propostos o modelo do presente trabalho. Este estudo deve-se estender também ao procedimento de planejamento de trajetória descrito no item 9.1 que apresenta uma sugestão da utilização do mapa construído na navegação robótica.
- Implementação do processo de localização, ou seja, utilizar a representação realizada pelo Autômato de Mapeamento para executar o procedimento que possibilita ao robô conhecer

seu próprio posicionamento quando inserido neste ambiente. Aliado ao sistema de navegação, deve-se implementar também um processo semelhante ao descrito no trabalho de Lim e Leonard (2000) chamado de re-localização (“relocation”), no qual o robô determina seu próprio posicionamento dentro de um ambiente, fundamentado nas medições feitas no local, no mapa disponível do ambiente e no descarte de localizações consideradas absurdas pelo cálculo baseado nas posições anteriores e em seus últimos deslocamentos.

- Mesmo com a construção de mapas precisos, o robô pode encontrar incoerências entre o espaço físico e sua representação, devido às características dinâmicas apresentadas pelos ambientes. Utilizando ações adaptativas similares às propostas no presente trabalho, pode-se realizar a atualização do mapa previamente construído sempre que o robô detectar mudanças no ambiente durante seu processo de navegação, executando-se um procedimento semelhante ao apresentado por Paris e LeCadre (2002).
- De acordo com a sugestão apresentada na pesquisa de Jennings, Watts e Tanis (1998), pode-se incluir na representação do mapa o registro de um índice de tempo que indica o histórico de alterações na configuração do ambiente a fim de identificar os obstáculos móveis presentes. Conforme tal identificação, há a possibilidade de se facilitar a representação de ambientes dinâmicos, registrando-se em diferentes autômatos os obstáculos fixos e os obstáculos móveis, empregando-se estratégia análoga à apresentada por Wolf e Shukhatme (2003), a qual mantém duas Grades de Ocupação distintas com o objetivo de mapear objetos móveis e imóveis. Pode-se, ainda, utilizar o registro de um

índice de ocupação nos estados livres com a finalidade de se identificar zonas regularmente ocupadas e assim evitar percursos próximos a locais com dinâmica intensa no processo de planejamento de trajetória.

- Devido à presente proposta não se restringir a nenhum tipo de sistemas de sensores, adotou-se como princípio, que o robô deve visitar todos os espaços livres do local a ser explorado. Após a definição do alcance da visibilidade do sistema sensor do agente robótico, as ações adaptativas apresentadas devem ser alteradas de modo a registrar os dados coletados de todo o espaço detectado pelo robô, e não somente as suas regiões adjacentes.
- Também após a definição do sistema sensor empregado no robô, pode-se acrescentar às representações do mapa construído um índice da probabilidade de ocupação nas regiões detectadas como obstáculos, devido aos erros que as medidas realizadas pelos sensores podem conter. Conforme o trabalho de pesquisa de Thrun (2002), a adoção de métodos de mapeamento que registrem informações de probabilidade é essencial às aplicações reais da robótica móvel.
- Empregando-se as características naturalmente oferecidas pelo formalismo adaptativo escolhido para a representação dos mapas, pode-se evoluir o sistema apresentado no presente trabalho de forma a equipar um time de robôs que possuam a tarefa de mapear um ambiente de maior extensão. Este processo é chamado de mapeamento distribuído e pode-se citar como exemplo de seu desenvolvimento as pesquisas de Aicardi (1995) e de

Jennings, Watts e Tanis (1998). Todo o time de robô deve utilizar autômatos idênticos para a realização do mapeamento, sistemas coerentes de definição das direções de orientação do mapa e se dispersarem pelo ambiente a ser mapeado. Quando dois robôs se encontram, executam o processo de identificação das partes comuns a cada um dos mapas construídos independentemente, por exemplo, pela localização dos marcos definidos nas porções dos espaços já mapeados, e unificam os dados obtidos, de tal modo a ampliar a representação completa do ambiente em questão.

- Finalmente, cita-se a argumentação de Brooks (1985) que, conforme o aumento da complexidade do agente robótico e da tarefa a ser desempenhada por ele no ambiente, o mapa construído deve conter informações das três dimensões do espaço. Kato, Murakami e Ohnishi (2002) apresentam um trabalho no qual a representação do local mapeado é realizada por um conjunto de cubos elementares classificados em vazios, ocupados e desconhecidos. Mapeamento semelhante pode ser realizado utilizando-se os Autômatos Adaptativos de forma a expandir o conjunto de estados para uma representação do espaço no qual o robô está imerso.

## 10. REFERÊNCIAS BIBLIOGRÁFICAS

ACAR, E.U.; CHOSET, H. Robust sensor-based coverage of unstructured environments. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Maui, EUA, 2001. **Proceedings**. 2001. p.61-68.

AICARDI, M. Coordination and control of a team of mobile robots. In: INRIA/IEEE SYMPOSIUM ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, Paris, França, 1995. **Proceedings**. 1995. p.335-344.

AOKI, T. et al. Acquisition of optimal action selection to avoid moving obstacles in autonomous mobile robot. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND AUTOMATION, Minneapolis, EUA, 1996. **Proceedings**. 1996. p.2055-2060.

BATALIN, M.A.; SUKHATME, G.S. Efficient exploration without localization. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Taipei, Taiwan, 2003. **Proceedings**. 2003. p.2714-2719.

BREZETZ, S.B. et al. Uncertain map making in natural environments. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND AUTOMATION, Minneapolis, EUA, 1996. **Proceedings**. 1996. p.1048-1053.

BROOKS, R.A. Visual map making for a mobile robot. In: IEEE INTERNATIONAL CONFERENCE IN ROBOTICS AND AUTOMATION, Saint Louis, EUA, 1985, **Proceedings**. 1985. p.824–829.

\_\_\_\_\_. A robust layered control system for a mobile robot. **IEEE Journal of Robotics and Automation**, v.2, n.1, p.14–23, 1986.

\_\_\_\_\_. Intelligence without reason. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, Sidney, Austrália, 1991. **Proceedings**. 1991. p.569–595.

BURLINA, P.; DEMENTHON, D.; DAVIS, L.S. Probabilistic navigation methods for uncertain and dynamic environments. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, The Hague, Holanda, 1992. **Proceedings**. 1992. p.312-315.

CARVALHO, R.N. et al. Complete coverage path planning and guidance for cleaning robots. In: INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, Guimarães, Portugal, 1997. **Proceedings**. 1997. p.677-682.

CHOW, H.N.; XU, Y.; TSO, S.K. Learning human navigational skill for smart wheelchair. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Lausanne, Suíça, 2002. **Proceedings**. 2002. p.996-1001.

COSTA, A.H.R. **Robótica móvel inteligente: progressos e desafios**. 2003. Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, 2003.

DRAKE, K.C.; MCVEY, E.S; IÑIGO, R.M. Experimental position and ranging results for a mobile robot. **IEEE Journal of Robotics and Automation**, v.3, n.1, p.31-42, 1987.

\_\_\_\_\_. Sensor roll angle error for a mobile robot using a navigation line. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.10, n.5, p.727-731, 1988.

ELFES, A. Sonar-based real-world mapping and navigation. **IEEE Journal of Robotics and Automation**, v.3, n.3, p.249-265, 1987.

FOK, K.Y; KABUKA, M.R. An automatic navigation system for vision guided vehicles using a double heuristic and a finite state machine. **IEEE Transactions on Robotics and Automation**, v.7, n.1, p.181-189, 1991.

FREITAS, A.V.; NETO, J.J. Uma ferramenta para construção de aplicações multilinguagens de programação. CONGRESSO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, El Calafate, Argentina, 2001. **Proceedings**. 2001.

FU, Y.; LANG, S.Y.T. Fuzzy logic based mobile robot area filling with vision system for indoor environments. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, Monterey, EUA, 1999. **Proceedings**. 1999. p.326-331.

GONZÁLEZ, E. et al. Complementary regions: a surface filling algorithm. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND AUTOMATION, Minneapolis, EUA, 1996. **Proceedings**. 1996. p.909-914.

GÜRTLER, C.; VAJTA, L.; NAGY, I. Trajectory planning for mobile robots based on dynamical models. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ENGINEERING SYSTEMS, Budapest, Hungria, 1997. **Proceedings**. 1997. p. 171-174.

HOWELL, J.; DONALD, B.R. Practical mobile robot self-localization. In: IEEE INTERNATIONAL CONFERENCE IN ROBOTICS AND AUTOMATION, San Francisco, EUA, 2000. **Proceedings**. 2000. p.3485-3492.

HWANG, K.S.; JU, M.Y. A propagating interface model strategy for global trajectory planning among moving obstacles. **IEEE Transactions on Industrial Electronics**, v.49, n.6, p.1313-1322, 2002.

HWANG, K.S. et al. Navigation based on sonar sensing in a partially structured environment. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS & SIGNAL PROCESSING, Nanjing, China, 2003, **Proceedings**. 2003. p.454-456.

INDELICATO, A. et al. A structure for the recognition of geometrical references for mobile robots using fuzzy logic. In: MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, Ames, EUA, 1997. **Proceedings**. 1997. p.565-568.

INNOCENTI, C. et al. Trajectory planning and real-time control of an autonomous mobile robot equipped with vision and ultrasonic sensors. In: IEEE/RSJ/GI INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Munique, Alemanha, 1994. **Proceedings**. 1994. p.1861-1866.

JENNINGS, J.; WATTS, C.K.; TANIS, C. Distributed map-making and navigation in dynamic environments. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Victoria, Canadá, 1998. **Proceedings**. 1998. p.1695-1701.

JÚNIOR, J.R.A.; NETO, J.J.; HIRAKAWA, A.R. Adaptive Automata for Independent Autonomous Navigation in Unknown Environment. In: INTERNATIONAL CONFERENCE ON APPLIED SIMULATION AND MODELLING, Banff, Alberta, 2000. **Anais**. 2000. p.317-322.

KATO, A.; MURAKAMI, T.; OHNISHI, K. Trajectory planning and environment recognition for unknown environment. In: INTERNATIONAL WORKSHOP ON ADVANCED MOTION CONTROL, Maribor, Eslovênia, 2002. **Proceedings**. 2002. p.578-582.

LEWIS, H. R.; PAPADIMITRIOU, C. H. **Elements of the theory of computation**. 1.ed. New Jersey: Prentice-Hall, Inc, 1981.

LIM, J.H.; LEONARD, J.J. Mobile robot relocation from echolocation constraints. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.22, n.9, p.1035-1041, 2000.

LORIGO, L.M.; BROOKS, R.A.; GRIMSON, W.E.L. Visually-guided obstacle avoidance in unstructured environments. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Grenoble, França, 1997. **Proceedings**. 1997. p.373-379

LUMELSKY, V.J.; MUKHOPADHYAY, S.; SUN, K. Dynamic path planning in sensor-based terrain acquisition. **IEEE Transactions on Robotics and Automation**, v.6, n.4, p.707-717, 1990.

LUO, C; YANG, S.X; MENG, M. Entire region filling in indoor environments using neural networks. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION, Shanghai, China, 2002. **Proceedings**. 2002. p.2039-2044.

MA, C.; LI, W.; LIU, L. Mobile robot motion by integration of low-level behavior control and high-level global planning. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, Beijing, China, 1996. **Proceedings**. 1996. p.310-315.

MARCO, M.D.; PRATTICCHIZZO, D.; VICINO, A. Polygonal approximation of uncertain height fields for outdoor navigation. In: IEEE CONFERENCE ON DECISION AND CONTROL, Sidney, Austrália, 2000. **Proceedings**. 2000., p.839-844.

MATTOS, L; GRANT, E. Passive sonar applications: target tracking and NAVIGATION OF AN AUTONOMOUS ROBOT. IN: IEEE INTERNATIONAL Conference in Robotics and Automation, New Orleans, EUA, 2004. **Proceedings**. 2004. p.4265-4270.

MECHLIH, H. Trajectory planning for mobile robots in a dynamic environment. In: Vehicle Navigation & Information Systems Conference, Ottawa, Canadá, 1993. **Proceedings.** 1993.p.551-554.

MENEZES, C.E.D.; NETO, J.J. Um método híbrido para a construção de etiquetadores morfológicos, aplicado à língua portuguesa, baseado em Autômatos Adaptativos. CONFERENCIA IBEROAMERICANA EN SISTEMAS, CIBERNÉTICA E INFORMÁTICA, Orlando, Florida, 2002. **Anais.** 2002.

MOVAREC, H.P.; ELFES, A. High resolution maps from wide angle sonar. In: IEEE INTERNATIONAL CONFERENCE IN ROBOTICS AND AUTOMATION, Saint Louis, EUA, 1985, **Proceedings.** 1985. p.116-121.

NETO, J. J. **Introdução à compilação.** Rio de Janeiro: Editora LTC, 1987.

\_\_\_\_\_. **Contribuições à metodologia de construção de compiladores.** 1993. Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.

\_\_\_\_\_. Adaptive Automata for Context-Dependent Languages. **ACM SIGPLAN Notices,** v.29, n.9, pp.115-24, 1994.

NETO, J.J.; MORAES, M. Formalismo adaptativo aplicado ao reconhecimento de linguagem natural. In: CONFERENCIA IBEROAMERICANA EN SISTEMAS, CIBERNÉTICA E INFORMÁTICA, Florida, EUA. 2002. **Anais**. 2002.

PARIS, S.; LE CADRE, J.P. Planification for Terrain-Aided Navigation. In: Fusion, Maryland, EUA, 2002. **Proceedings**. 2002. pp. 1007-1014.

PISTORI, H.; NETO, J.J. AdapTree - proposta de um algoritmo para indução de árvores de decisão baseado em técnicas adaptativas. In: CONFERÊNCIA LATINO AMERICANA DE INFORMÁTICA, Montevideú, Uruguai, 2002. **Anais**. 2002.

PRADO, J.P.A.; ABE, J.M. Paraconsistent automata and robot's behavior. In: CONGRESS OF LOGIC APPLIED TO TECHNOLOGY, Marília, Brasil, 2003. **Anais**. 2003.

RAO, N.S.V. et al. On terrain acquisition by a finite-sized mobile robot in plane. In: IEEE INTERNATIONAL CONFERENCE IN ROBOTICS AND AUTOMATION, Raleigh, EUA, 1987. **Proceedings**. 1987. p.1314-1319.

SCHMIDT, G.; HOFNER, C. An advanced planning and navigation approach for autonomous cleaning robot operations. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, Victoria, Canadá, 1998. **Proceedings**. 1998. p.1230-1235.

SMIERZCHALSKI, R. An intelligent method of ship's trajectory planning at sea. In: IEEE/IEEJ/JSAI INTERNATIONAL CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, Tóquio, Japão, 1999. **Proceedings.** 1999. p.907-912.

SOUSA, M.A.A.; HIRAKAWA, A.R.; NETO, J.J. Adaptive Automata for Mapping Unknown Environments by Mobile Robots. In: IBERO-AMERICAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, Puebla, México, 2004a. **Lecture Notes in Artificial Intelligence: Advances in Artificial Intelligence.** 2004a. p.562-571.

\_\_\_\_\_. Adaptive Automata for Mobile Robotic Mapping. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL NEURAL NETWORKS, São Luís, Brasil, 2004b. **Proceedings.** 2004b. CD-ROM.

SOUSA, M.A.A.; HIRAKAWA, A.R. Robotic mapping and navigation in unknown environments using adaptive automata. In: INTERNATIONAL CONFERENCE ON ADAPTIVE AND NATURAL COMPUTING ALGORITHMS, Coimbra, Portugal, 2005. **Proceedings.** 2005.

THRUN, S. Robotic mapping: a survey. In: LAKEMEYER, G.; NEBEL, B. **Exploring Artificial Intelligence in the New Millenium.** EUA: Morgan Kaufmann, 2002. p.1-29.

TSUMURA, T. Survey of automated guided vehicle in Japanese factory. In: IEEE INTERNATIONAL CONFERENCE IN ROBOTICS AND AUTOMATION, San Francisco, EUA, 1986. **Proceedings.** 1986. p.1329-1334.

WOLF, D.F; SUKHATME, G.S. Towards mapping dynamic environments. In: INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, Coimbra, Portugal, 2003. **Proceedings**. 2003. p.594-600.

YAMAUCHI, B. A frontier-based approach for autonomous exploration. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, Monterey, EUA, 1997. **Proceedings**. 1997. p.146-151.

YASUTOMI, F. et al. Cleaning robot control. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Philadelphia, EUA, 1988. **Proceedings**. 1988. p.1839-1841.

ZELINSKY, A. A mobile robot exploration algorithm. **IEEE Transactions on Robotics and Automation**, v.8, n.6, p.707-717, 1992.

ZHU, S.Q.; LEWIS, F.L. Trajectory Planning and robust stabilization for auto-drive cars. In: AMERICAN CONTROL CONFERENCE, Seattle, EUA, 1995. **Proceedings**. 1995. p.820-824.

ZIMMER, U.R. Embedding local metrical map patches in a globally consistent topological map. In: INTERNATIONAL SYMPOSIUM ON UNDERWATER TECHNOLOGY, Tóquio, Japão, 2000. **Proceedings**. 2000. p.301-305.

## Apêndice A

### Representação Algébrica da Configuração Inicial do Autômato de Mapeamento

A seguir, apresenta-se a configuração inicial do Autômato de Mapeamento.

$s = 40$ $F = \{ 2 \}$ $P = \{$	$(3, b) : B(3) \rightarrow 6,$ $(3, i) : I(3) \rightarrow 6,$ $(6, a) : A(6) \rightarrow 3,$ $(6, h) : H(6) \rightarrow 3,$ $(4, b) : B(4) \rightarrow 7,$ $(4, i) : I(4) \rightarrow 7,$ $(7, a) : A(7) \rightarrow 4,$ $(7, h) : H(7) \rightarrow 4,$ $(5, b) : B(5) \rightarrow 8,$ $(5, i) : I(5) \rightarrow 8,$ $(8, a) : A(8) \rightarrow 5,$ $(8, h) : H(8) \rightarrow 5,$ $(6, b) : B(6) \rightarrow 9,$ $(6, i) : I(6) \rightarrow 9,$ $(9, a) : A(9) \rightarrow 6,$ $(9, h) : H(9) \rightarrow 6,$ $(10, b) : B(10) \rightarrow 1,$ $(10, i) : I(10) \rightarrow 1,$ $(1, a) : A(1) \rightarrow 10,$ $(1, h) : H(1) \rightarrow 10,$ $(3, c) : C(3) \rightarrow 11,$ $(3, j) : J(3) \rightarrow 11,$ $(11, d) : D(11) \rightarrow 3,$ $(11, k) : K(11) \rightarrow 3,$ $(12, c) : C(12) \rightarrow 7,$ $(12, j) : J(12) \rightarrow 7,$ $(7, d) : D(7) \rightarrow 12,$ $(7, k) : K(7) \rightarrow 12,$ $(9, b) : B(9) \rightarrow 13,$ $(9, i) : I(9) \rightarrow 13,$ $(13, a) : A(13) \rightarrow 9,$ $(13, h) : H(13) \rightarrow 9,$	$(40, \#) : AA \rightarrow 41,$ $(41, \#) : BB \rightarrow 42,$ $(42, \#) : CC \rightarrow 43,$ $(43, \#) : DD \rightarrow 44,$ $(44, \#) : V \rightarrow 5,$ $(1, r) : R(1) \rightarrow 1,$ $(2, r) : R(2) \rightarrow 2,$ $(3, r) : R(3) \rightarrow 3,$ $(4, r) : R(4) \rightarrow 4,$ $(5, r) : R(5) \rightarrow 5,$ $(6, r) : R(6) \rightarrow 6,$ $(7, r) : R(7) \rightarrow 7,$ $(8, r) : R(8) \rightarrow 8,$ $(9, r) : R(9) \rightarrow 9,$ $(10, r) : R(10) \rightarrow 10,$ $(11, r) : R(11) \rightarrow 11,$ $(12, r) : R(12) \rightarrow 12,$ $(13, r) : R(13) \rightarrow 13,$ $(1, m) : M(1) \rightarrow 1,$ $(2, m) : M(2) \rightarrow 2,$ $(3, m) : M(3) \rightarrow 3,$ $(4, m) : M(4) \rightarrow 4,$ $(5, m) : M(5) \rightarrow 5,$ $(6, m) : M(6) \rightarrow 6,$ $(7, m) : M(7) \rightarrow 7,$ $(8, m) : M(8) \rightarrow 8,$ $(9, m) : M(9) \rightarrow 9,$ $(10, m) : M(10) \rightarrow 10,$ $(11, m) : M(11) \rightarrow 11,$ $(12, m) : M(12) \rightarrow 12,$ $(13, m) : M(13) \rightarrow 13,$ $(1, f) \rightarrow 1,$ $(3, p) \rightarrow 3,$ $(7, q) \rightarrow 7,$ $(9, g) \rightarrow 9,$ $(20, a) \rightarrow 21,$ $(24, b) \rightarrow 25,$ $(28, c) \rightarrow 29,$ $(32, d) \rightarrow 33\}$
$(1, c) : C(1) \rightarrow 2,$ $(1, j) : J(1) \rightarrow 2,$ $(2, d) : D(2) \rightarrow 1,$ $(2, k) : K(2) \rightarrow 1,$ $(2, c) : C(2) \rightarrow 3,$ $(2, j) : J(2) \rightarrow 3,$ $(3, d) : D(3) \rightarrow 2,$ $(3, k) : K(3) \rightarrow 2,$ $(4, c) : C(4) \rightarrow 5,$ $(4, j) : J(4) \rightarrow 5,$ $(5, d) : D(5) \rightarrow 4,$ $(5, k) : K(5) \rightarrow 4,$ $(5, c) : C(5) \rightarrow 6,$ $(5, j) : J(5) \rightarrow 6,$ $(6, d) : D(6) \rightarrow 5,$ $(6, k) : K(6) \rightarrow 5,$ $(7, c) : C(7) \rightarrow 8,$ $(7, j) : J(7) \rightarrow 8,$ $(8, d) : D(8) \rightarrow 7,$ $(8, k) : K(8) \rightarrow 7,$ $(8, c) : C(8) \rightarrow 9,$ $(8, j) : J(8) \rightarrow 9,$ $(9, d) : D(9) \rightarrow 8,$ $(9, k) : K(9) \rightarrow 8,$ $(1, b) : B(1) \rightarrow 4,$ $(1, i) : I(1) \rightarrow 4,$ $(4, a) : A(4) \rightarrow 1,$ $(4, h) : H(4) \rightarrow 1,$ $(2, b) : B(2) \rightarrow 5,$ $(2, i) : I(2) \rightarrow 5,$ $(5, a) : A(5) \rightarrow 2,$ $(5, h) : H(5) \rightarrow 2,$		

A seguir, apresentam-se as ações adaptativas do Autômato de Mapeamento.

```

A(e){var1:
-[(e,a):A->var1]
-[(e,h):H->var1]
-[(var1,b):B->e]
-[(var1,i):I->e]
+[(var1,b)->var1]
+[(e,n):NN->var1:FF]
+[(var1,s):SS->e:GG]
+[(e,a)->e]}

I(e){var1:
-[(e,b):B->var1]
-[(e,i):I->var1]
-[(var1,a):A->e]
-[(var1,h):H->e]
+[(var1,h)->var1]
+[(e,x)->var1]
+[(var1,w)->e]
+[(e,i)->e]}

B(e){var1:
-[(e,b):B->var1]
-[(e,i):I->var1]
-[(var1,a):A->e]
-[(var1,h):H->e]
+[(var1,a)->var1]
+[(e,s):SS->var1:GG]
+[(var1,n):NN->e:FF]
+[(e,b)->e]}

J(e){var1:
-[(e,c):C->var1]
-[(e,j):J->var1]
-[(var1,d):D->e]
-[(var1,k):K->e]
+[(var1,k)->var1]
+[(e,y)->var1]
+[(var1,z)->e]
+[(e,j)->e]}

C(e){var1,var2,var3:
-[(e,c):C->var1]
-[(e,j):J->var1]
-[(var1,d):D->e]
-[(var1,k):K->e]
+[(var1,d)->var1]
+[(e,l):LL->var1:PP]
+[(var1,o):OO->e:QQ]
+[(e,c)->e]}

K(e){var1:
-[(e,d):D->var1]
-[(e,k):K->var1]
-[(var1,c):C->e]
-[(var1,j):J->e]
+[(var1,j)->var1]
+[(e,z)->var1]
+[(var1,y)->e]
+[(e,k)->e]}

D(e){var1,var2,var3:
-[(e,d):D->var1]
-[(e,k):K->var1]
-[(var1,c):C->e]
-[(var1,j):J->e]
+[(var1,c)->var1]
+[(e,o):OO->var1:QQ]
+[(var1,l):LL->e:PP]
+[(e,d)->e]}

NN{var1,var2:
?[(var1,f)->var1]
?[(23,var2)->22]
N(var1,var2)}

N(e,ee){var1,var2,ger1*:
?[(e,a):A->var1]
?[(e,c):C->var2]
+[(ger1,b):B(ger1)->var2]
+[(ger1,i):I(ger1)->var2]
+[(var2,a):A(var2)->ger1]
+[(var2,h):H(var2)->ger1]
+[(var1,c):C(var1)->ger1]
+[(var1,j):J(var1)->ger1]
+[(ger1,d):D(ger1)->var1]
+[(ger1,k):K(ger1)->var1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
N(var2,ee)}

H(e){var1:
-[(e,a):A->var1]
-[(e,h):H->var1]
-[(var1,b):B->e]
-[(var1,i):I->e]
+[(var1,i)->var1]
+[(e,w)->var1]
+[(var1,x)->e]
+[(e,h)->e]}

```

```
SS{var1,var2:
?[(var1,g)->var1]
?[(27,var2)->26]
S(var1,var2)}
```

```
S(e,ee){var1,var2,ger1*:
?[(e,b):B->var1]
?[(e,d):D->var2]
+[(var2,b):B(var2)->ger1]
+[(var2,i):I(var2)->ger1]
+[(ger1,a):A(ger1)->var2]
+[(ger1,h):H(ger1)->var2]
+[(ger1,c):C(ger1)->var1]
+[(ger1,j):J(ger1)->var1]
+[(var1,d):D(var1)->ger1]
+[(var1,k):K(var1)->ger1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
S(var2,ee)}
```

```
LL{var1,var2:
?[(var1,p)->var1]
?[(31,var2)->30]
L(var1,var2)}
```

```
L(e,ee){var1,var2,ger1*:
?[(e,c):C->var1]
?[(e,b):B->var2]
+[(var2,c):C(var2)->ger1]
+[(var2,j):J(var2)->ger1]
+[(ger1,d):D(ger1)->var2]
+[(ger1,k):K(ger1)->var2]
+[(var1,b):B(var1)->ger1]
+[(var1,i):I(var1)->ger1]
+[(ger1,a):A(ger1)->var1]
+[(ger1,h):H(ger1)->var1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
L(var2,ee)}
```

```
OO{var1,var2:
?[(var1,q)->var1]
?[(35,var2)->34]
O(var1,var2)}
```

```
O(e,ee){var1,var2,ger1*:
?[(e,d):D->var1]
?[(e,a):A->var2]
+[(ger1,c):C(ger1)->var2]
+[(ger1,j):J(ger1)->var2]
+[(var2,d):D(var2)->ger1]
+[(var2,k):K(var2)->ger1]
+[(ger1,b):B(ger1)->var1]
+[(ger1,i):I(ger1)->var1]
+[(var1,a):A(var1)->ger1]
+[(var1,h):H(var1)->ger1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
O(var2,ee)}
```

```
FF{var1:
?[(23,var1)->22]
F(var1)}
```

```
F(e){var1,var2,var3,var4,var5,
var6,var7,var8,var16,ger1*:
-[(var16,f)->var16]
?[(var16,a):A->var1]
+[(var1,f)->var1]
+[(ger1,b):B(ger1)->var1]
+[(ger1,i):I(ger1)->var1]
+[(var1,a):A(var1)->ger1]
+[(var1,h):H(var1)->ger1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
?[(var4,p)->var4]
?[(var4,c):C->var5]
?[(var4,a):A->var6]
-[(var4,c):C->var5]
-[(var4,j):J->var5]
-[(var5,d):D->var4]
-[(var5,k):K->var4]
-[(var7,b):B->var5]
-[(var7,i):I->var5]
-[(var5,a):A->var7]
-[(var5,h):H->var7]
-[(var7,c):C->var8]
-[(var7,j):J->var8]
-[(var8,d):D->var7]
-[(var8,k):K->var7]
-[(var5,r):R->var5]
-[(var8,r):R->var8]
-[(var5,m):M->var5]
-[(var8,m):M->var8]
-[(var4,p)->var4]
+[(var6,p)->var6]}
```

```
GG{var1:
?[(27,var1)->26]
G(var1)}
```

```
G(e){var1,var2,var3,var4,var5,
var6,var7,var8,var16,ger1*:
-[(var16,g)->var16]
?[(var16,b):B->var1]
+[(var1,g)->var1]
+[(var1,b):B(var1)->ger1]
+[(var1,i):I(var1)->ger1]
+[(ger1,a):A(ger1)->var1]
+[(ger1,h):H(ger1)->var1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
?[(var4,q)->var4]
?[(var4,d):D->var5]
?[(var4,b):B->var6]
-[(var5,c):C->var4]
-[(var5,j):J->var4]
-[(var4,d):D->var5]
-[(var4,k):K->var5]
-[(var5,b):B->var7]
-[(var5,i):I->var7]
-[(var7,a):A->var5]
-[(var7,h):H->var5]
-[(var8,c):C->var7]
-[(var8,j):J->var7]
-[(var7,d):D->var8]
-[(var7,k):K->var8]
-[(var5,r):R->var5]
-[(var8,r):R->var8]
-[(var5,m):M->var5]
-[(var8,m):M->var8]
-[(var4,q)->var4]
+[(var6,q)->var6]}
```

```
PP{var1:
?[(31,var1)->30]
P(var1)}
```

```
P(e){var1,var2,var3,var4,var5,
var6,var7,var16,ger1*:
-[(var16,p)->var16]
?[(var16,c):C->var1]
+[(var1,p)->var1]
+[(var1,c):C(var1)->ger1]
+[(var1,j):J(var1)->ger1]
+[(ger1,d):D(ger1)->var1]
+[(ger1,k):K(ger1)->var1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
?[(var4,g)->var4]
?[(var4,b):B->var5]
?[(var4,c):C->var6]
-[(var4,b):B->var5]
-[(var4,i):I->var5]
-[(var5,a):A->var4]
-[(var5,h):H->var4]
-[(var5,c):C->var7]
-[(var5,j):J->var7]
-[(var7,d):D->var5]
-[(var7,k):K->var5]
-[(var7,b):B->var8]
-[(var7,i):I->var8]
-[(var8,a):A->var7]
-[(var8,h):H->var7]
-[(var5,r):R->var5]
-[(var8,r):R->var8]
-[(var5,m):M->var5]
-[(var8,m):M->var8]
-[(var4,g)->var4]
+[(var6,g)->var6]}
```

```
QQ{var1:
?[(35,var1)->34]
Q(var1)}
```

```

Q(e){var1,var2,var3,var4,var5,
var6,var7,var16,ger1*:
-[(var16,q)->var16]
?[(var16,d):D->var1]
+[(var1,q)->var1]
+[(ger1,c):C(ger1)->var1]
+[(ger1,j):J(ger1)->var1]
+[(var1,d):D(var1)->ger1]
+[(var1,k):K(var1)->ger1]
+[(ger1,r):R(ger1)->ger1]
+[(ger1,m):M(ger1)->ger1]
?[(var4,f)->var4]
?[(var4,a):A->var5]
?[(var4,d):D->var6]
-[(var5,b):B->var4]
-[(var5,i):I->var4]
-[(var4,a):A->var5]
-[(var4,h):H->var5]
-[(var7,c):C->var5]
-[(var7,j):J->var5]
-[(var5,d):D->var7]
-[(var5,k):K->var7]
-[(var8,b):B->var7]
-[(var8,i):I->var7]
-[(var7,a):A->var8]
-[(var7,h):H->var8]
-[(var5,r):R->var5]
-[(var8,r):R->var8]
-[(var5,m):M->var5]
-[(var8,m):M->var8]
-[(var4,f)->var4]
+[(var6,f)->var6]}

```

```

M(e){var1:
-[(44,#):V->var1]
+[(44,#):V->e]
-[(23,a)->22]
-[(27,a)->26]
-[(31,a)->30]
-[(35,a)->34]
T(e,e,e,e)}

```

```

T(e,ee,uu,u)
{var1,var2,var3,var4,var5:
?[(e,a):A->var1]
?[(ee,n):NN->var2:FF]
?[(uu,w)->var3]
+[(15,a)->var1]
+[(15,n)->var2]
+[(15,w)->var3]
?[(15,var4)->var5]
+[(20,var4)->21]
-[(15,var4)->var5]
TT(var5,var5,var5,u)}

```

```

TT(e,ee,uu,u)
{var1,var2,var3,var4,var5:
?[(e,a):A->var1]
?[(ee,n):NN->var2:FF]
?[(uu,w)->var3]
+[(15,a)->var1]
+[(15,n)->var2]
+[(15,w)->var3]
?[(15,var4)->var5]
+[(22,var4)->23]
-[(15,var4)->var5]
U(u,u,u,u)}

```

```

U(e,ee,uu,u)
{var1,var2,var3,var4,var5:
?[(e,b):B->var1]
?[(ee,s):SS->var2:GG]
?[(uu,x)->var3]
+[(15,b)->var1]
+[(15,s)->var2]
+[(15,x)->var3]
?[(15,var4)->var5]
+[(24,var4)->25]
-[(15,var4)->var5]
UU(var5,var5,var5,u)}

```

```

UU(e,ee,uu,u)
{var1,var2,var3,var4,var5:
?[(e,b):B->var1]
?[(ee,s):SS->var2:GG]
?[(uu,x)->var3]
+[(15,b)->var1]
+[(15,s)->var2]
+[(15,x)->var3]
?[(15,var4)->var5]
+[(26,var4)->27]
-[(15,var4)->var5]
X(u,u,u,u)}

```

```

X(e, ee, uu, u)
{var1, var2, var3, var4, var5:
?[(e, c):C->var1]
?[(ee, l):LL->var2:PP]
?[(uu, y)->var3]
+[(15, c)->var1]
+[(15, l)->var2]
+[(15, y)->var3]
?[(15, var4)->var5]
+[(28, var4)->29]
-[(15, var4)->var5]
XX(var5, var5, var5, u)}

XX(e, ee, uu, u)
{var1, var2, var3, var4, var5:
?[(e, c):C->var1]
?[(ee, l):LL->var2:PP]
?[(uu, y)->var3]
+[(15, c)->var1]
+[(15, l)->var2]
+[(15, y)->var3]
?[(15, var4)->var5]
+[(30, var4)->31]
-[(15, var4)->var5]
Z(u, u, u, u)}

Z(e, ee, uu, u)
{var1, var2, var3, var4, var5:
?[(e, d):D->var1]
?[(ee, o):OO->var2:QQ]
?[(uu, z)->var3]
+[(15, d)->var1]
+[(15, o)->var2]
+[(15, z)->var3]
?[(15, var4)->var5]
+[(32, var4)->33]
-[(15, var4)->var5]
ZZ(var5, var5, var5, u)
}

ZZ(e, ee, uu, u)
{var1, var2, var3, var4, var5:
?[(e, d):D->var1]
?[(ee, o):OO->var2:QQ]
?[(uu, z)->var3]
+[(15, d)->var1]
+[(15, o)->var2]
+[(15, z)->var3]
?[(15, var4)->var5]
+[(34, var4)->35]
-[(15, var4)->var5]}

AA{var1:
?[(22, var1)->23]
+[(23, a)->22]
HH(var1)}

HH(var1){:
-[(23, a)->22]}

BB{var1:
?[(26, var1)->27]
+[(27, a)->26]
II(var1)}

II(var1){:
-[(27, a)->26]}

CC{var1:
?[(30, var1)->31]
+[(31, a)->30]
JJ(var1)}

JJ(var1){:
-[(31, a)->30]}

DD{var1:
?[(34, var1)->35]
+[(35, a)->34]
KK(var1)}

KK(var1){:
-[(35, a)->34]}

V{var1, var2, var3, var4, var5, var6,
var7, var8:
-[(20, var1)->21]
-[(22, var2)->23]
-[(24, var3)->25]
-[(26, var4)->27]
-[(28, var5)->29]
-[(30, var6)->31]
-[(32, var7)->33]
-[(34, var8)->35]}

R(e){:
+[(45, r)->e]}

```

## Apêndice B

### Representação Algébrica da Configuração Inicial do Autômato de Exploração

A seguir, apresenta-se a configuração inicial do Autômato de Exploração.

s = 5	(623,v) -> 624,	(706,z) -> 707,
F = { 30 }	(624,#) :X->29,	(706,o) -> 710,
P={	(608,m) -> 625,	(706,d) -> 718,
	(625,g) :B->630,	(707,o) -> 708,
(1,#) -> 2,	(625,t) -> 626,	(707,z) -> 708,
(3,#) -> 4,	(626,p) :C->633,	(707,d) -> 708,
(5,#) :Y->6,	(633,q) -> 634,	(707,e) -> 708,
(22,a) -> 21,	(633,v) -> 634,	(710,o) -> 708,
	(626,u) -> 627,	(710,z) -> 708,
(6,n) -> 601,	(627,q) :K->628,	(710,d) -> 711,
(6,w) -> 601,	(627,v) -> 628,	(710,e) -> 711,
(6,a) -> 601,	(630,p) -> 631:T,	(709,c) -> 712,
(601,n) -> 602,	(630,u) -> 631:S,	(709,e) -> 712,
(601,w) -> 602,	(631,q) -> 632:V,	(712,z) -> 713,
(601,a) -> 602,	(631,v) -> 632:U,	(712,o) -> 715,
(601,e) -> 602,	(628,#) :X->29,	(712,d) -> 719,
(602,s) -> 603,	(632,#) :X->29,	(713,o) -> 714,
(602,x) -> 603,	(634,#) :X->29,	(713,z) -> 714,
(602,b) -> 603,		(713,d) -> 714,
(603,s) -> 604,	(7,#) -> 700,	(713,e) -> 714,
(603,x) -> 604,	(700,n) -> 701,	(715,o) -> 714,
(603,b) -> 604,	(700,w) -> 701,	(715,z) -> 714,
(603,e) -> 604,	(700,a) -> 701,	(715,d) -> 716,
(604,l) -> 605,	(701,n) -> 702,	(715,e) -> 716,
(604,y) -> 605,	(701,w) -> 702,	(717,l) -> 712,
(604,c) -> 605,	(701,a) -> 702,	(717,y) -> 712,
(605,l) -> 606,	(701,e) -> 702,	(717,c) -> 712,
(605,y) -> 606,	(702,s) -> 703,	(717,e) -> 712,
(605,c) -> 606,	(702,x) -> 703,	(718,o) -> 711,
(605,e) -> 606,	(702,b) -> 703,	(718,z) -> 711,
(606,o) -> 607,	(703,s) -> 704,	(718,d) -> 711,
(606,z) -> 607,	(703,x) -> 704,	(718,e) -> 711,
(606,d) -> 607,	(703,b) -> 704,	(719,o) -> 716,
(607,o) -> 608,	(703,e) -> 704,	(719,z) -> 716,
(607,z) -> 608,	(704,y) -> 705,	(719,d) -> 716,
(607,d) -> 608,	(704,l) -> 709,	(719,e) -> 716,
(607,e) -> 608,	(704,c) -> 717,	(708,f) -> 721,
(608,f) :A->621,	(705,l) -> 706,	(708,m) -> 721,
(621,g) -> 622,	(705,y) -> 706,	(721,g) :B->722,
(621,t) -> 622,	(705,c) -> 706,	(721,t) :F->722,
(622,p) -> 623,	(705,e) -> 706,	(722,p) -> 723:T,
(622,u) -> 623,	(709,l) -> 706,	(722,u) -> 723:S,
(623,q) -> 624,	(709,y) -> 706,	(723,q) -> 724:V,

(723,v)->724:U,  
 (724,#):X->29,  
 (711,f)->731,  
 (711,m)->731,  
 (731,g):B->732,  
 (732,p)->733:T,  
 (732,u)->733:S,  
 (733,q)->734:V,  
 (733,v)->734:U,  
 (731,t)->735,  
 (735,p)->736,  
 (735,u)->736,  
 (736,q):M->737:V,  
 (736,v):FS->737:P,  
 (734,#):X->29,  
 (737,#):X->28,  
 (714,f)->741,  
 (714,m)->741,  
 (741,g):B->742,  
 (742,p)->743:T,  
 (742,u)->743:S,  
 (743,q)->744:V,  
 (743,v)->744:U,  
 (741,t)->745,  
 (745,p):C->746:T,  
 (746,q)->747:V,  
 (746,v)->747:U,  
 (745,u):FS->748:P,  
 (748,v)->749,  
 (748,q)->749,  
 (744,#):X->29,  
 (747,#):X->28,  
 (749,#):X->29,  
 (716,f)->751,  
 (716,m)->751,  
 (751,g):B->752,  
 (752,p)->753:T,  
 (752,u)->753:S,  
 (753,q)->754:V,  
 (753,v)->754:U,  
 (751,t)->755,  
 (755,p):C->756:T,  
 (756,q)->757:V,  
 (756,v)->757:U,  
 (755,u)->758,  
 (758,q):M->757:V,  
 (758,v):FS->759:P,  
 (754,#):X->29,  
 (757,#):X->28,  
 (759,#):X->29,  
 (8,n)->801,  
 (8,w)->801,  
 (8,a)->801,  
 (801,n)->802,  
 (801,w)->802,  
 (801,a)->802,  
 (801,e)->802,  
 (802,s)->803,  
 (802,x)->803,  
 (802,b)->803,  
 (803,s)->804,  
 (803,x)->804,  
 (803,b)->804,  
 (803,e)->804,  
 (804,y)->805,  
 (804,l)->809,  
 (804,c)->817,  
 (805,l)->806,  
 (805,y)->806,  
 (805,c)->806,  
 (805,e)->806,  
 (809,l)->806,  
 (809,y)->806,  
 (806,z)->807,  
 (806,o)->810,  
 (806,d)->818,  
 (807,o)->808,  
 (807,z)->808,  
 (807,d)->808,  
 (807,e)->808,  
 (810,o)->808,  
 (810,z)->808,  
 (810,d)->811,  
 (810,e)->811,  
 (809,c)->812,  
 (809,e)->812,  
 (812,z)->813,  
 (812,o)->815,  
 (812,d)->819,  
 (813,o)->814,  
 (813,z)->814,  
 (813,d)->814,  
 (813,e)->814,  
 (815,o)->814,  
 (815,z)->814,  
 (815,d)->816,  
 (815,e)->816,  
 (817,l)->812,  
 (817,y)->812,  
 (817,c)->812,  
 (817,e)->812,  
 (818,o)->811,  
 (818,z)->811,  
 (818,d)->811,  
 (818,e)->811,  
 (819,o)->816,  
 (819,z)->816,  
 (819,d)->816,  
 (819,e)->816,  
 (808,f)->821,  
 (821,g):N->822,  
 (821,t):H->826,  
 (822,p)->823,  
 (822,u)->823,  
 (823,q)->824,  
 (823,v)->824,  
 (808,m)->825,  
 (825,g):B->826,  
 (825,t):F->826,  
 (826,p)->827:T,  
 (826,u)->827:S,  
 (827,q)->828:V,  
 (827,v)->828:U,  
 (824,#):X->29,  
 (828,#):X->29,  
 (811,f)->831,  
 (831,g):N->832,  
 (831,t):H->836,  
 (832,p)->833,  
 (832,u)->833,  
 (833,q)->834,  
 (833,v)->834,  
 (811,m)->835,  
 (835,g):B->836,  
 (835,t):F->833,  
 (836,p)->837:T,  
 (836,u)->837:S,  
 (837,q)->838:V,  
 (837,v)->838:U,  
 (834,#):X->29,  
 (838,#):X->29,  
 (814,f)->841,  
 (841,g):N->842,  
 (841,t):H->847,  
 (842,p)->843,  
 (842,u)->843,  
 (843,q)->844,  
 (843,v)->844,  
 (814,m)->845,  
 (845,g):B->847,  
 (845,t)->846,  
 (846,p):C->850:T,  
 (846,u):F->843,  
 (850,q)->840:V,

(850,v)->840:U,  
 (847,p)->848:T,  
 (847,u)->848:S,  
 (848,q)->849:V,  
 (848,v)->849:U,  
 (840,#):X->28,  
 (844,#):X->29,  
 (849,#):X->29,  
 (816,f)->851,  
 (851,g):N->852,  
 (851,t):H->857,  
 (852,p)->853,  
 (852,u)->853,  
 (853,q)->854,  
 (853,v)->854,  
 (816,m)->855,  
 (855,g):B->857,  
 (855,t)->856,  
 (856,p):C->860:T,  
 (856,u):F->853,  
 (860,q)->861:V,  
 (860,v)->861:U,  
 (857,p)->858:T,  
 (857,u)->858:S,  
 (858,q)->859:V,  
 (858,v)->859:U,  
 (854,#):X->29,  
 (859,#):X->29,  
 (861,#):X->28,  
  
 (9,#)->900,  
 (900,n)->901,  
 (900,w)->901,  
 (900,a)->901,  
 (901,n)->902,  
 (901,w)->902,  
 (901,a)->902,  
 (901,e)->902,  
 (902,s)->903,  
 (902,x)->903,  
 (902,b)->903,  
 (903,s)->904,  
 (903,x)->904,  
 (903,b)->904,  
 (903,e)->904,  
 (904,y)->905,  
 (904,l)->909,  
 (904,c)->917,  
 (905,l)->906,  
 (905,y)->906,  
 (905,c)->906,  
 (905,e)->906,  
  
 (909,l)->906,  
 (909,y)->906,  
 (906,z)->907,  
 (906,o)->910,  
 (906,d)->918,  
 (907,o)->908,  
 (907,z)->908,  
 (907,d)->908,  
 (907,e)->908,  
 (910,o)->908,  
 (910,z)->908,  
 (910,d)->911,  
 (910,e)->911,  
 (909,c)->912,  
 (909,e)->912,  
 (912,z)->913,  
 (912,o)->915,  
 (912,d)->919,  
 (913,o)->914,  
 (913,z)->914,  
 (913,d)->914,  
 (913,e)->914,  
 (915,o)->914,  
 (915,z)->914,  
 (915,d)->916,  
 (915,e)->916,  
 (917,l)->912,  
 (917,y)->912,  
 (917,c)->912,  
 (917,e)->912,  
 (918,o)->911,  
 (918,z)->911,  
 (918,d)->911,  
 (918,e)->911,  
 (919,o)->916,  
 (919,z)->916,  
 (919,d)->916,  
 (919,e)->916,  
 (908,f):H->921,  
 (908,m):F->921,  
 (921,g)->922,  
 (921,t)->922,  
 (922,p)->923:T,  
 (922,u)->923:S,  
 (923,q)->924:V,  
 (923,v)->924:U,  
 (924,#):X->29,  
 (911,f):H->931,  
 (931,g)->932,  
 (931,t)->932,  
 (932,p)->933:T,  
 (932,u)->933:S,  
  
 (933,q)->934:V,  
 (933,v)->934:U,  
 (911,m)->935,  
 (935,g)->936,  
 (935,t)->936,  
 (936,p)->937,  
 (936,u)->937,  
 (937,q):K->938:V,  
 (937,v):FN->938:P,  
 (934,#):X->29,  
 (938,#):X->28,  
 (914,f):H->941,  
 (941,g)->942,  
 (941,t)->942,  
 (942,p)->943:T,  
 (942,u)->943:S,  
 (943,q)->944:V,  
 (943,v)->944:U,  
 (914,m)->945,  
 (945,g)->946,  
 (945,t)->946,  
 (946,p):I->947:T,  
 (947,q)->948:V,  
 (947,v)->948:U,  
 (946,u):FN->949:P,  
 (949,q)->950,  
 (949,v)->950,  
 (944,#):X->29,  
 (948,#):X->28,  
 (950,#):X->29,  
 (916,f):H->951,  
 (951,g)->952,  
 (951,t)->952,  
 (952,p)->953:T,  
 (952,u)->953:S,  
 (953,q)->954:V,  
 (953,v)->954:U,  
 (916,m)->955,  
 (955,g)->956,  
 (955,t)->956,  
 (956,p):I->959:T,  
 (959,q)->960:V,  
 (959,v)->960:U,  
 (956,u)->957,  
 (957,q):K->960:V,  
 (957,v):FN->958:P,  
 (954,#):X->29,  
 (958,#):X->29,  
 (960,#):X->28,

```

(10,n)->1001,
(10,w)->1001,
(10,a)->1001,
(1001,n)->1002,
(1001,w)->1002,
(1001,a)->1002,
(1001,e)->1002,
(1002,s)->1003,
(1002,x)->1003,
(1002,b)->1003,
(1003,s)->1004,
(1003,x)->1004,
(1003,b)->1004,
(1003,e)->1004,
(1004,y)->1005,
(1004,l)->1009,
(1004,c)->1017,
(1005,l)->1006,
(1005,y)->1006,
(1005,c)->1006,
(1005,e)->1006,
(1009,l)->1006,
(1009,y)->1006,
(1006,z)->1007,
(1006,o)->1010,
(1006,d)->1018,
(1007,o)->1008,
(1007,z)->1008,
(1007,d)->1008,
(1007,e)->1008,
(1010,o)->1008,
(1010,z)->1008,
(1010,d)->1011,
(1010,e)->1011,
(1009,c)->1012,
(1009,e)->1012,
(1012,z)->1013,
(1012,o)->1015,
(1012,d)->1019,
(1013,o)->1014,
(1013,z)->1014,
(1013,d)->1014,
(1013,e)->1014,
(1015,o)->1014,
(1015,z)->1014,
(1015,d)->1016,
(1015,e)->1016,
(1017,l)->1012,
(1017,y)->1012,
(1017,c)->1012,
(1017,e)->1012,
(1018,o)->1011,
(1018,z)->1011,
(1018,d)->1011,
(1018,e)->1011,
(1019,o)->1016,
(1019,z)->1016,
(1019,d)->1016,
(1019,e)->1016,
(1008,f)->1021,
(1021,g):O->1022,
(1021,t):H->1026,
(1022,p)->1023,
(1022,u)->1023,
(1023,q)->1024,
(1023,v)->1024,
(1008,m)->1025,
(1025,g):B->1026,
(1025,t):F->1023,
(1026,p)->1027:T,
(1026,u)->1027:S,
(1027,q)->1028:V,
(1027,v)->1028:U,
(1024,#):X->29,
(1028,#):X->29,
(1011,f)->1031,
(1031,g):O->1032,
(1031,t):H->1036,
(1032,p)->1033,
(1032,u)->1033,
(1033,q)->1034,
(1033,v)->1034,
(1011,m)->1035,
(1035,g):B->1036,
(1035,t):F->1033,
(1036,p)->1037:T,
(1036,u)->1037:S,
(1037,q)->1038:V,
(1037,v)->1038:U,
(1034,#):X->29,
(1038,#):X->29,
(1014,f)->1041,
(1041,g):O->1042,
(1041,t):H->1047,
(1042,p)->1043,
(1042,u)->1043,
(1043,q)->1044,
(1043,v)->1044,
(1014,m)->1045,
(1045,g):B->1047,
(1047,p)->1048:T,
(1047,u)->1048:S,
(1048,q)->1049:V,
(1048,v)->1049:U,
(1045,t)->1046,
(1046,p):I->1050:T,
(1046,u):F->1043,
(1050,q)->1040:V,
(1050,v)->1040:U,
(1040,#):X->28,
(1044,#):X->29,
(1049,#):X->29,
(1016,f)->1051,
(1051,g):O->1052,
(1051,t):H->1057,
(1052,p)->1053,
(1052,u)->1053,
(1053,q)->1054,
(1053,v)->1054,
(1016,m)->1055,
(1055,g):B->1057,
(1055,t)->1056,
(1056,p):I->1060:T,
(1056,u):F->1053,
(1060,q)->1061:V,
(1060,v)->1061:U,
(1057,p)->1058:T,
(1057,u)->1058:S,
(1058,q)->1059:V,
(1058,v)->1059:U,
(1054,#):X->29,
(1059,#):X->29,
(1061,#):X->28,
(11,#)->1100,
(1100,n)->1101,
(1100,w)->1101,
(1100,a)->1101,
(1101,n)->1102,
(1101,w)->1102,
(1101,a)->1102,
(1101,e)->1102,
(1102,s)->1103,
(1102,x)->1103,
(1102,b)->1103,
(1103,s)->1104,
(1103,x)->1104,
(1103,b)->1104,
(1103,e)->1104,
(1104,y)->1105,
(1104,l)->1109,
(1104,c)->1117,
(1105,l)->1106,
(1105,y)->1106,
(1105,c)->1106,
(1105,e)->1106,

```

(1109,l)->1106,  
 (1109,y)->1106,  
 (1106,z)->1107,  
 (1106,o)->1110,  
 (1106,d)->1118,  
 (1107,o)->1108,  
 (1107,z)->1108,  
 (1107,d)->1108,  
 (1107,e)->1108,  
 (1110,o)->1108,  
 (1110,z)->1108,  
 (1110,d)->1111,  
 (1110,e)->1111,  
 (1109,c)->1112,  
 (1109,e)->1112,  
 (1112,z)->1113,  
 (1112,o)->1115,  
 (1112,d)->1119,  
 (1113,o)->1114,  
 (1113,z)->1114,  
 (1113,d)->1114,  
 (1113,e)->1114,  
 (1115,o)->1114,  
 (1115,z)->1114,  
 (1115,d)->1116,  
 (1115,e)->1116,  
 (1117,l)->1112,  
 (1117,y)->1112,  
 (1117,c)->1112,  
 (1117,e)->1112,  
 (1118,o)->1111,  
 (1118,z)->1111,  
 (1118,d)->1111,  
 (1118,e)->1111,  
 (1119,o)->1116,  
 (1119,z)->1116,  
 (1119,d)->1116,  
 (1119,e)->1116,  
 (1108,f):J->1121,  
 (1108,m):G->1125,  
 (1121,g)->1122,  
 (1121,t)->1122,  
 (1122,p)->1123:T,  
 (1122,u)->1123:S,  
 (1123,q)->1124:V,  
 (1123,v)->1124:U,  
 (1125,g)->1126,  
 (1125,t)->1126,  
 (1126,p)->1127:T,  
 (1126,u)->1127:S,  
 (1127,q)->1128:V,  
 (1127,v)->1128:U,  
 (1124,#):X->29,  
 (1128,#):X->29,  
 (1111,f):J->1131,  
 (1131,g)->1132,  
 (1131,t)->1132,  
 (1132,p)->1133:T,  
 (1132,u)->1133:S,  
 (1133,q)->1134:V,  
 (1133,v)->1134:U,  
 (1111,m)->1135,  
 (1135,g)->1136,  
 (1135,t)->1136,  
 (1136,p)->1137,  
 (1136,u)->1137,  
 (1137,q):K->1138:V,  
 (1137,v):GN->1138:P,  
 (1134,#):X->29,  
 (1138,#):X->28,  
 (1114,f):J->1141,  
 (1141,g)->1142,  
 (1141,t)->1142,  
 (1142,p)->1143:T,  
 (1142,u)->1143:S,  
 (1143,q)->1144:V,  
 (1143,v)->1144:U,  
 (1114,m)->1145,  
 (1145,g)->1146,  
 (1145,t)->1146,  
 (1146,p):I->1147:T,  
 (1147,q)->1148:V,  
 (1147,v)->1148:U,  
 (1146,u):GN->1149:P,  
 (1149,q)->1150,  
 (1149,v)->1150,  
 (1144,#):X->29,  
 (1148,#):X->28,  
 (1150,#):X->29,  
 (1116,f):J->1151,  
 (1151,g)->1152,  
 (1151,t)->1152,  
 (1152,p)->1153:T,  
 (1152,u)->1153:S,  
 (1153,q)->1154:V,  
 (1153,v)->1154:U,  
 (1116,m)->1155,  
 (1155,g)->1156,  
 (1155,t)->1156,  
 (1156,p)->1157:T,  
 (1157,q):K->1158:V,  
 (1157,v):I->1158:U,  
 (1156,u)->1159,  
 (1159,q):K->1158:V,  
 (1159,v):GN->1160:P,  
 (1154,#):X->29,  
 (1158,#):X->28,  
 (1160,#):X->29,  
 (12,n)->1201,  
 (12,w)->1201,  
 (12,a)->1201,  
 (1201,n)->1202,  
 (1201,w)->1202,  
 (1201,a)->1202,  
 (1201,e)->1202,  
 (1202,s)->1203,  
 (1202,x)->1203,  
 (1202,b)->1203,  
 (1203,s)->1204,  
 (1203,x)->1204,  
 (1203,b)->1204,  
 (1203,e)->1204,  
 (1204,y)->1205,  
 (1204,l)->1209,  
 (1204,c)->1217,  
 (1205,l)->1206,  
 (1205,y)->1206,  
 (1205,c)->1206,  
 (1205,e)->1206,  
 (1209,l)->1206,  
 (1209,y)->1206,  
 (1206,z)->1207,  
 (1206,o)->1210,  
 (1206,d)->1218,  
 (1207,o)->1208,  
 (1207,z)->1208,  
 (1207,d)->1208,  
 (1207,e)->1208,  
 (1210,o)->1208,  
 (1210,z)->1208,  
 (1210,d)->1211,  
 (1210,e)->1211,  
 (1209,c)->1212,  
 (1209,e)->1212,  
 (1212,z)->1213,  
 (1212,o)->1215,  
 (1212,d)->1219,  
 (1213,o)->1214,  
 (1213,z)->1214,  
 (1213,d)->1214,  
 (1213,e)->1214,  
 (1215,o)->1214,  
 (1215,z)->1214,  
 (1215,d)->1216,  
 (1215,e)->1216,

(1217,l)->1212,  
 (1217,y)->1212,  
 (1217,c)->1212,  
 (1217,e)->1212,  
 (1218,o)->1211,  
 (1218,z)->1211,  
 (1218,d)->1211,  
 (1218,e)->1211,  
 (1219,o)->1216,  
 (1219,z)->1216,  
 (1219,d)->1216,  
 (1219,e)->1216,  
 (1208,f)->1221,  
 (1221,g):Q->1222,  
 (1221,t):J->1222,  
 (1222,p)->1223,  
 (1222,u)->1223,  
 (1223,q)->1224,  
 (1223,v)->1224,  
 (1208,m)->1225,  
 (1225,g):L->1226,  
 (1225,t):G->1226,  
 (1226,p)->1227:T,  
 (1226,u)->1227:S,  
 (1227,q)->1228:V,  
 (1227,v)->1228:U,  
 (1224,#):X->29,  
 (1228,#):X->29,  
 (1211,f)->1231,  
 (1231,g):Q->1232,  
 (1231,t):J->1239,  
 (1232,p)->1233,  
 (1232,u)->1233,  
 (1233,q)->1234,  
 (1233,v)->1234,  
 (1211,m)->1235,  
 (1235,g):L->1239,  
 (1235,t)->1236,  
 (1236,p)->1237:T,  
 (1236,u)->1237:S,  
 (1237,q):K->1238:V,  
 (1237,v):G->1238,  
 (1239,p)->1240:T,  
 (1239,u)->1240:S,  
 (1240,q)->1230:V,  
 (1240,v)->1230:U,  
 (1234,#):X->29,  
 (1238,#):X->28,  
 (1230,#):X->29,  
 (1214,f)->1241,  
 (1241,g):Q->1242,  
 (1241,t):J->1246,  
 (1242,p)->1243,  
 (1242,u)->1243,  
 (1243,q)->1244,  
 (1243,v)->1244,  
 (1214,m)->1245,  
 (1245,g):L->1246,  
 (1245,t):F->1246,  
 (1246,p)->1247:T,  
 (1246,u)->1247:S,  
 (1247,q)->1248:V,  
 (1247,v)->1248:U,  
 (1244,#):X->29,  
 (1248,#):X->29,  
 (1216,f)->1251,  
 (1251,g):Q->1252,  
 (1251,t):J->1259,  
 (1252,p)->1253,  
 (1252,u)->1253,  
 (1253,q)->1254,  
 (1253,v)->1254,  
 (1216,m)->1255,  
 (1255,g):L->1259,  
 (1255,t)->1256,  
 (1256,p)->1257:T,  
 (1256,u)->1257:S,  
 (1257,q):K->1258:V,  
 (1257,v):G->1258,  
 (1259,p)->1260:T,  
 (1259,u)->1260:S,  
 (1260,q)->1261:V,  
 (1260,v)->1261:U,  
 (1254,#):X->29,  
 (1258,#):X->28,  
 (1261,#):X->29,  
 (13,#)->1300,  
 (1300,n)->1301,  
 (1300,w)->1301,  
 (1300,a)->1301,  
 (1301,n)->1302,  
 (1301,w)->1302,  
 (1301,a)->1302,  
 (1301,e)->1302,  
 (1302,s)->1303,  
 (1302,x)->1303,  
 (1302,b)->1303,  
 (1303,s)->1304,  
 (1303,x)->1304,  
 (1303,b)->1304,  
 (1303,e)->1304,  
 (1304,y)->1305,  
 (1304,l)->1309,  
 (1304,c)->1317,  
 (1305,l)->1306,  
 (1305,y)->1306,  
 (1305,c)->1306,  
 (1305,e)->1306,  
 (1309,l)->1306,  
 (1309,y)->1306,  
 (1306,z)->1307,  
 (1306,o)->1310,  
 (1306,d)->1318,  
 (1307,o)->1308,  
 (1307,z)->1308,  
 (1307,d)->1308,  
 (1307,e)->1308,  
 (1310,o)->1308,  
 (1310,z)->1308,  
 (1310,d)->1311,  
 (1310,e)->1311,  
 (1309,c)->1312,  
 (1309,e)->1312,  
 (1312,z)->1313,  
 (1312,o)->1315,  
 (1312,d)->1319,  
 (1313,o)->1314,  
 (1313,z)->1314,  
 (1313,d)->1314,  
 (1313,d)->1314,  
 (1315,o)->1314,  
 (1315,z)->1314,  
 (1315,d)->1316,  
 (1315,e)->1316,  
 (1317,l)->1312,  
 (1317,y)->1312,  
 (1317,c)->1312,  
 (1317,e)->1312,  
 (1318,o)->1311,  
 (1318,z)->1311,  
 (1318,d)->1311,  
 (1318,e)->1311,  
 (1319,o)->1316,  
 (1319,z)->1316,  
 (1319,d)->1316,  
 (1319,e)->1316,  
 (1308,f)->1321,  
 (1308,m)->1321,  
 (1321,g):L->1322,  
 (1321,t):G->1322,  
 (1322,p)->1323:T,  
 (1322,u)->1323:S,  
 (1323,q)->1324:V,  
 (1323,v)->1324:U,  
 (1324,#):X->29,

(1311,f)->1331,  
 (1311,m)->1331,  
 (1331,g):L->1332,  
 (1332,p)->1333:T,  
 (1332,u)->1333:S,  
 (1333,q)->1334:V,  
 (1333,v)->1334:U,  
 (1331,t)->1335,  
 (1335,p)->1336,  
 (1335,u)->1336,  
 (1336,q):M->1337,  
 (1336,v):GS->1334:P,  
 (1334,#):X->29,  
 (1337,#):X->28,  
 (1314,f)->1341,  
 (1314,m)->1341,  
 (1341,g):L->1342,  
 (1342,p)->1343:T,  
 (1342,u)->1343:S,  
 (1343,q)->1344:V,  
 (1343,v)->1344:U,  
 (1341,t)->1345,  
 (1345,p):C->1346:T,  
 (1346,q)->1347:V,  
 (1346,v)->1347:U,  
 (1345,u):GS->1348:P,  
 (1348,q)->1349,  
 (1348,v)->1349,  
 (1344,#):X->29,  
 (1347,#):X->28,  
 (1349,#):X->29,  
 (1316,f)->1351,  
 (1316,m)->1351,  
 (1351,g):L->1352,  
 (1352,p)->1353:T,  
 (1352,u)->1353:S,  
 (1353,q)->1354:V,  
 (1353,v)->1354:U,  
 (1351,t)->1355,  
 (1355,p)->1356:T,  
 (1356,q):M->1357:V,  
 (1356,v):C->1357:U,  
 (1355,u)->1358,  
 (1358,q):M->1357:V,  
 (1358,v):GS->1359:P,  
 (1354,#):X->29,  
 (1357,#):X->28,  
 (1359,#):X->29,  
  
 (14,n)->1401,  
 (14,w)->1401,  
 (14,a)->1401,  
  
 (1401,n)->1402,  
 (1401,w)->1402,  
 (1401,a)->1402,  
 (1401,e)->1402,  
 (1402,s)->1403,  
 (1402,x)->1403,  
 (1402,b)->1403,  
 (1403,s)->1404,  
 (1403,x)->1404,  
 (1403,b)->1404,  
 (1403,e)->1404,  
 (1404,y)->1405,  
 (1404,l)->1409,  
 (1404,c)->1417,  
 (1405,l)->1406,  
 (1405,y)->1406,  
 (1405,c)->1406,  
 (1405,e)->1406,  
 (1409,l)->1406,  
 (1409,y)->1406,  
 (1406,z)->1407,  
 (1406,o)->1410,  
 (1406,d)->1418,  
 (1407,o)->1408,  
 (1407,z)->1408,  
 (1407,d)->1408,  
 (1407,e)->1408,  
 (1410,o)->1408,  
 (1410,z)->1408,  
 (1410,d)->1411,  
 (1410,e)->1411,  
 (1409,c)->1412,  
 (1409,e)->1412,  
 (1412,z)->1413,  
 (1412,o)->1415,  
 (1412,d)->1419,  
 (1413,o)->1414,  
 (1413,z)->1414,  
 (1413,d)->1414,  
 (1413,e)->1414,  
 (1415,o)->1414,  
 (1415,z)->1414,  
 (1415,d)->1416,  
 (1415,e)->1416,  
 (1417,l)->1412,  
 (1417,y)->1412,  
 (1417,c)->1412,  
 (1417,e)->1412,  
 (1418,o)->1411,  
 (1418,z)->1411,  
 (1418,d)->1411,  
 (1418,e)->1411,  
  
 (1419,o)->1416,  
 (1419,z)->1416,  
 (1419,d)->1416,  
 (1419,e)->1416,  
 (1408,f)->1421,  
 (1421,g):R->1422,  
 (1421,t):J->1426,  
 (1422,p)->1423,  
 (1422,u)->1423,  
 (1423,q)->1424,  
 (1423,v)->1424,  
 (1408,m)->1425,  
 (1425,g):L->1426,  
 (1425,t):G->1426,  
 (1426,p)->1427:T,  
 (1426,u)->1427:S,  
 (1427,q)->1428:V,  
 (1427,v)->1428:U,  
 (1424,#):X->29,  
 (1428,#):X->29,  
 (1411,f)->1431,  
 (1431,g):R->1432,  
 (1431,t):J->1439,  
 (1432,p)->1433,  
 (1432,u)->1433,  
 (1433,q)->1434,  
 (1433,v)->1434,  
 (1411,m)->1435,  
 (1435,g):L->1439,  
 (1435,t)->1436,  
 (1436,p)->1437:T,  
 (1436,u)->1437:S,  
 (1437,q):M->1438:V,  
 (1437,v):G->1430,  
 (1439,p)->1440:T,  
 (1439,u)->1440:S,  
 (1440,q)->1430:V,  
 (1440,v)->1430:U,  
 (1434,#):X->29,  
 (1438,#):X->28,  
 (1430,#):X->29,  
 (1414,f)->1441,  
 (1441,g):R->1442,  
 (1441,t):J->1446,  
 (1442,p)->1443,  
 (1442,u)->1443,  
 (1443,q)->1444,  
 (1443,v)->1444,  
 (1414,m)->1445,  
 (1445,g):L->1446,  
 (1445,t):G->1446,  
 (1446,p)->1447:T,

(1446,u)->1447:S,  
 (1447,q)->1448:V,  
 (1447,v)->1448:U,  
 (1444,#):X->29,  
 (1448,#):X->29,  
 (1416,f)->1451,  
 (1451,g):R->1452,  
 (1451,t):J->1461,  
 (1452,p)->1453,  
 (1452,u)->1453,  
 (1453,q)->1454,  
 (1453,v)->1454,  
 (1416,m)->1455,  
 (1455,g):L->1461,  
 (1455,t)->1456,  
 (1456,p)->1457:T,  
 (1457,q):M->1458:V,  
 (1457,v):F->1460,  
 (1456,u)->1459,  
 (1459,g):M->1458:V,  
 (1459,t):G->1460,  
 (1461,p)->1462:T,  
 (1461,u)->1462:S,  
 (1462,q)->1463:V,  
 (1462,v)->1463:U,  
 (1454,#):X->29,  
 (1458,#):X->28,  
 (1460,#):X->29,  
 (1463,#):X->29,  
  
 (15,n)->1501,  
 (15,w)->1501,  
 (15,a)->1501,  
 (1501,n)->1502,  
 (1501,w)->1502,  
 (1501,a)->1502,  
 (1501,e)->1502,  
 (1502,s)->1503,  
 (1502,x)->1503,  
 (1502,b)->1503,  
 (1503,s)->1504,  
 (1503,x)->1504,  
 (1503,b)->1504,  
 (1503,e)->1504,  
 (1504,y)->1505,  
 (1504,l)->1509,  
 (1504,c)->1517,  
 (1505,l)->1506,  
 (1505,y)->1506,  
 (1505,c)->1506,  
 (1505,e)->1506,  
 (1509,l)->1506,  
  
 (1509,y)->1506,  
 (1506,z)->1507,  
 (1506,o)->1510,  
 (1506,d)->1518,  
 (1507,o)->1508,  
 (1507,z)->1508,  
 (1507,d)->1508,  
 (1507,e)->1508,  
 (1510,o)->1508,  
 (1510,z)->1508,  
 (1510,d)->1511,  
 (1510,e)->1511,  
 (1509,c)->1512,  
 (1509,e)->1512,  
 (1512,z)->1513,  
 (1512,o)->1515,  
 (1512,d)->1519,  
 (1513,o)->1514,  
 (1513,z)->1514,  
 (1513,d)->1514,  
 (1513,e)->1514,  
 (1515,o)->1514,  
 (1515,z)->1514,  
 (1515,d)->1516,  
 (1515,e)->1516,  
 (1517,l)->1512,  
 (1517,y)->1512,  
 (1517,c)->1512,  
 (1517,e)->1512,  
 (1518,o)->1511,  
 (1518,z)->1511,  
 (1518,d)->1511,  
 (1518,e)->1511,  
 (1519,o)->1516,  
 (1519,z)->1516,  
 (1519,d)->1516,  
 (1519,e)->1516,  
 (1508,f)->1521,  
 (1508,m)->1521,  
 (1521,g)->1522,  
 (1521,t)->1522,  
 (1522,p)->1523,  
 (1522,u)->1523,  
 (1523,q):F->1524,  
 (1523,v):F->1524,  
 (1511,f)->1531,  
 (1511,m)->1531,  
 (1531,g)->1532,  
 (1531,t)->1532,  
 (1532,p)->1533,  
 (1532,u)->1533,  
 (1533,q):K->1534:D,  
  
 (1533,v):F->1534,  
 (1514,f)->1541,  
 (1514,m)->1541,  
 (1541,g)->1542,  
 (1541,t)->1542,  
 (1542,p):C->1543:D,  
 (1542,u):F->1543,  
 (1543,q)->1544,  
 (1543,v)->1544,  
 (1516,f)->1551,  
 (1516,m)->1551,  
 (1551,g)->1552,  
 (1551,t)->1552,  
 (1552,p):C->1553:D,  
 (1553,q)->1554,  
 (1553,v)->1554,  
 (1552,u)->1555,  
 (1555,q):K->1556:D,  
 (1555,v):F->1556,  
  
 (16,n)->1601,  
 (16,w)->1601,  
 (16,a)->1601,  
 (1601,n)->1602,  
 (1601,w)->1602,  
 (1601,a)->1602,  
 (1601,e)->1602,  
 (1602,s)->1603,  
 (1602,x)->1603,  
 (1602,b)->1603,  
 (1603,s)->1604,  
 (1603,x)->1604,  
 (1603,b)->1604,  
 (1603,e)->1604,  
 (1604,y)->1605,  
 (1604,l)->1609,  
 (1604,c)->1617,  
 (1605,l)->1606,  
 (1605,y)->1606,  
 (1605,c)->1606,  
 (1605,e)->1606,  
 (1609,l)->1606,  
 (1609,y)->1606,  
 (1606,z)->1607,  
 (1606,o)->1610,  
 (1606,d)->1618,  
 (1607,o)->1608,  
 (1607,z)->1608,  
 (1607,d)->1608,  
 (1607,e)->1608,  
 (1610,o)->1608,  
 (1610,z)->1608,

```

(1610,d)->1611,
(1610,e)->1611,
(1609,c)->1612,
(1609,e)->1612,
(1612,z)->1613,
(1612,o)->1615,
(1612,d)->1619,
(1613,o)->1614,
(1613,z)->1614,
(1613,d)->1614,
(1613,e)->1614,
(1615,o)->1614,
(1615,z)->1614,
(1615,d)->1616,
(1615,e)->1616,
(1617,l)->1612,
(1617,y)->1612,
(1617,c)->1612,
(1617,e)->1612,
(1618,o)->1611,
(1618,z)->1611,
(1618,d)->1611,

(1618,e)->1611,
(1619,o)->1616,
(1619,z)->1616,
(1619,d)->1616,
(1619,e)->1616,
(1608,f)->1621,
(1608,m)->1621,
(1621,g)->1622,
(1621,t)->1622,
(1622,p)->1623,
(1622,u)->1623,
(1623,q):G->1624,
(1623,v):G->1624,
(1611,f)->1631,
(1611,m)->1631,
(1631,g)->1632,
(1631,t)->1632,
(1632,p)->1633,
(1632,u)->1633,
(1633,q):K->1634:D,
(1633,v):G->1634,
(1614,f)->1641,

(1614,m)->1641,
(1641,g)->1642,
(1641,t)->1642,
(1642,p):C->1643:D,
(1642,u):G->1643,
(1643,q)->1644,
(1643,v)->1644,
(1616,f)->1651,
(1616,m)->1651,
(1651,g)->1652,
(1651,t)->1652,
(1652,p)->1653,
(1653,q):K->1654:D,
(1653,v):C->1654:D,
(1652,u)->1655,
(1655,q):K->1656:D,
(1655,v):F->1656,

(28,#):D->29,
(29,k):Z->30:W}

```

A seguir, apresentam-se as ações adaptativas do Autômato de Exploração.

```

A{var1,var2:
-[(1,var1)->2]
+[(1,n)->2]
-[(5,#):Y->var2]
+[(5,#):Y->6]}

B{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,s)->2]
-[(5,#):Y->var2]
+[(5,#):Y->7]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,n)->4]}

C{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,l)->2]
-[(5,#):Y->var2]
+[(5,#):Y->8]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,o)->4]}

F{var1,var2,var3,
var4,var5,var6,var7,
var8,var9,var10:
-[(1,var1)->2]
?[(var2,var3)->4]
?[(var4,a)->21]
?[(var4,var5)->var2]
-[(var6,var7)->4]
-[(var8,var9)->var6]
+[(var8,var9)->4]
+[(1,var7)->2]
-[(5,#):Y->var10]
+[(5,#):Y->15]
FF(var5)}

FF(e){var1,var2,
var3,var4,var5:
+[(2,r)->1]
-[(var1,a)->21]
-[(var1,#)->var2]
-[(var3,a)->var1]
+[(var3,a)->21]
?[(var4,a)->21]
?[(var4,var5)->4]
FFF(var5)}

FFF(e){var1,var2,
var3,var4,var5:
-[(var1,a)->21]
-[(var1,#)->4]
-[(var2,a)->var1]
+[(var2,a)->21]
?[(var4,a)->21]
?[(var4,var5)->4]
FFF(var5)}

G{var1,var2,var3,
var4,var5,var6,var7,
var8,var9,var10:
-[(1,var1)->2]
?[(var2,var3)->4]
?[(var4,a)->21]
?[(var4,var5)->var2]
-[(var6,var7)->4]
-[(var8,var9)->var6]
+[(var8,var9)->4]
+[(1,var7)->2]
-[(5,#):Y->var10]
+[(5,#):Y->16]
GG(var5)}

GG(e){var1,var2,
var3,var4,var5:
+[(2,r)->1]
-[(var1,a)->21]
-[(var1,#)->var2]
-[(var3,a)->var1]
+[(var3,a)->21]
?[(var4,a)->21]
?[(var4,var5)->4]
GGG(var5)}

GGG(e){var1,var2,
var3,var4,var5:
-[(var1,a)->21]
-[(var1,#)->4]
-[(var2,a)->var1]
+[(var2,a)->21]
?[(var4,a)->21]
?[(var4,var5)->4]
GGG(var5)}

H{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,n)->2]
-[(5,#):Y->var2]
+[(5,#):Y->9]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,s)->4]}

I{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,l)->2]
-[(5,#):Y->var2]
+[(5,#):Y->10]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,o)->4]}

J{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,n)->2]
-[(5,#):Y->var2]
+[(5,#):Y->11]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,s)->4]}

K{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,o)->2]
-[(5,#):Y->var2]
+[(5,#):Y->12]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,l)->4]}

```

```

L{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,s)->2]
-[(5,#):Y->var2]
+[(5,#):Y->13]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,n)->4]}

M{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,o)->2]
-[(5,#):Y->var2]
+[(5,#):Y->14]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,l)->4]}

N{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,s)->2]
-[(5,#):Y->var2]
+[(5,#):Y->8]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,n)->4]}

O{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,n)->2]
-[(5,#):Y->var2]
+[(5,#):Y->10]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,s)->4]}

Q{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,n)->2]
-[(5,#):Y->var2]
+[(5,#):Y->12]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,s)->4]}

R{var1,var2,var3,
var4,ger1*:
-[(1,var1)->2]
+[(1,s)->2]
-[(5,#):Y->var2]
+[(5,#):Y->14]
-[(var3,var4)->4]
+[(var3,var4)->ger1]
+[(ger1,n)->4]}

S{var1,var2:
?[(17,l)->var2]
-[(17,var1)->18]
+[(17,o)->18]
SS(var2)}

SS(i){var1,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->23]}

T{var1,var2:
?[(17,o)->var2]
-[(17,var1)->18]
+[(17,l)->18]
TT(var2)}

TT(i){var1,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->24]}

U{var1,var2:
?[(19,l)->var2]
-[(19,var1)->20]
+[(19,o)->20]
UU(var2)}

UU(i){var1,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->23]}

V{var1,var2:
?[(19,o)->var2]
-[(19,var1)->20]
+[(19,l)->20]
VV(var2)}

VV(i){var1,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->24]}

X{var1,var2,var3,
var4,var5,var6,var7:
?[(var2,var3)->4]
?[(var4,var5)->var2]
-[(var1,#)->23]
+[(var1,#)->var4]
-[(var6,#)->24]
+[(var6,#)->var2]
?[(25,var7)->26]
XX(var1,var2,var4,
var7)}

XX(i,j,k,l){var1,
var2,ger1*,ger2*:
?[(26,var1)->25]
-[(i,#)->k]
-[(k,var2)->j]
+[(k,var1)->ger1]
+[(ger1,var2)->ger2]
+[(ger2,l)->j]
+[(i,#)->ger2]}

Z{var1,var2,var3:
-[(25,var2)->26]
-[(26,var3)->25]
?[(var1,s)->4]
ZZ(var1)}

ZZ(i)
{var1,var2,var3:
?[(var1,var2)->i]
?[(var3,n)->var1]
ZZZ(i,var1,var3)}

```

```

ZZZ(i,j,k){var1,
var2,var4,var5:
+[(i,h)->i]
+[(j,h)->j]
?[(var1,a)->21]
?[(var1,#)->var2]
+[(var2,h)->var2]
-[(var2,h)->var2]
?[(i,var4)->i]
?[(j,var5)->j]
-[(i,h)->i]
-[(j,h)->j]
ZZZZ(i,j,k,
var4,var5)}

ZZZZ(i,j,k,l,m)
{var1,var2,var3:
-[(i,s)->4]
-[(j,var1)->i]
+[(j,var1)->4]
-[(k,n)->j]
-[(var2,var3)->k]
+[(var2,var3)->j]
+[(25,s)->26]
+[(26,n)->25]}

W{var1:
?[(var1,n)->4]
WW(var1)}

WW(i)
{var1,var2,var3:
?[(var1,var2)->i]
?[(var3,s)->var1]
WWW(i,var1,var3)}

WWW(i,j,k){var1,
var2,var4,var5:
+[(i,h)->i]
+[(j,h)->j]
?[(var1,a)->21]
?[(var1,#)->var2]
+[(var2,h)->var2]
-[(var2,h)->var2]
?[(i,var4)->i]
?[(j,var5)->j]
-[(i,h)->i]
-[(j,h)->j]
WWWW(i,j,k,
var4,var5)}

WWWW(i,j,k,l,m)
{var1,var2,var3:
-[(i,n)->4]
-[(j,var1)->i]
+[(j,var1)->4]
-[(k,s)->j]
-[(var2,var3)->k]
+[(var2,var3)->j]
+[(25,n)->26]
+[(26,s)->25]}

Y{:
-[(2,r)->1]}

FSS(i){var1,var2,
var4,var5,var6,
ger1*,ger2*:
-[(var1,#)->24]
-[(var1,a)->21]
-[(var4,a)->var1]
+[(var4,a)->21]
-[(1,var2)->2]
+[(1,n)->2]
+[(2,r)->1]
-[(var5,var6)->4]
+[(var5,var6)->ger1]
+[(ger1,s)->ger2]
+[(ger2,o)->4]}

FN{var1,var2,var3,
var4,var5,var6,var7,
var8:
?[(17,l)->var2]
-[(17,var1)->18]
+[(17,o)->18]
-[(var3,var4)->4]
-[(var5,var6)->var3]
+[(var5,var6)->4]
+[(1,var4)->2]
-[(5,#):Y->var7]
+[(5,#):Y->15]
FFN(var2)}

FFN(i)
{var1,var2,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->24]
?[(1,o)->var2]
FNN(var2)}

FNN(i){var1,var2,
var4,var5,var6,
ger1*,ger2*:
-[(var1,#)->24]
-[(var1,a)->21]
-[(var4,a)->var1]
+[(var4,a)->21]
-[(1,var2)->2]
+[(1,s)->2]
+[(2,r)->1]
-[(var5,var6)->4]
+[(var5,var6)->ger1]
+[(ger1,n)->ger2]
+[(ger2,o)->4]}

```

```

GS{var1,var2,var3,
var4,var5,var6,var7,
var8:
?[(19,1)->var2]
-[(19,var1)->20]
+[(19,o)->20]
-[(var3,var4)->4]
-[(var5,var6)->var3]
+[(var5,var6)->4]
+[(1,var4)->2]
-[(5,#):Y->var7]
+[(5,#):Y->16]
GGS(var2)}

```

```

GGS(i)
{var1,var2,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->24]
?[(1,1)->var2]
GSS(var2)}

```

```

GSS(i){var1,var2,
var4,var5,var6,
ger1*,ger2*:
-[(var1,#)->24]
-[(var1,a)->21]
-[(var4,a)->var1]
+[(var4,a)->21]
-[(1,var2)->2]
+[(1,n)->2]
+[(2,r)->1]
-[(var5,var6)->4]
+[(var5,var6)->ger1]
+[(ger1,s)->ger2]
+[(ger2,l)->4]}

```

```

GN{var1,var2,var3,
var4,var5,var6,var7,
var8:
?[(19,1)->var2]
-[(19,var1)->20]
+[(19,o)->20]
-[(var3,var4)->4]
-[(var5,var6)->var3]
+[(var5,var6)->4]
+[(1,var4)->2]
-[(5,#):Y->var7]
+[(5,#):Y->16]
GGN(var2)}

```

```

GGN(i)
{var1,var2,ger1*:
-[(var1,a)->21]
+[(var1,a)->ger1]
+[(ger1,a)->21]
+[(ger1,#)->24]
?[(1,1)->var2]
GNN(var2)}

```

```

GNN(i){var1,var2,
var4,var5,var6,
ger1*,ger2*:
-[(var1,#)->24]
-[(var1,a)->21]
-[(var4,a)->var1]
+[(var4,a)->21]
-[(1,var2)->2]
+[(1,s)->2]
+[(2,r)->1]
-[(var5,var6)->4]
+[(var5,var6)->ger1]
+[(ger1,n)->ger2]
+[(ger2,l)->4]}

```

```

P{var1,var2:
?[(var1,a)->21]
?[(var1,var2)->4]
PP(var2)}

```

```

PP(i)
{var1,var2,var3:
+[(2,r)->1]
-[(var1,a)->21]
-[(var1,#)->4]
-[(var2,a)->var1]
+[(var2,a)->21]

```

```

?[(var2,var3)->4]
PP(var3)}

```

```

D{var1,var2,var3:
-[(17,var1)->18]
-[(19,var2)->20]
?[(1,var3)->2]}

```

## Anexo A

### Descrição formal dos autômatos finitos determinísticos

Seja o autômato finito determinístico  $M = \{Q, \Sigma, \delta, q_0, F\}$

com:

$Q$ : conjunto finito de estados;

$\Sigma$ : alfabeto de entrada;

$\delta$ : função de transição de estados  $\delta = Q \times \Sigma \rightarrow Q$ ;

$q_0$ : estado inicial único  $q_0 \in Q$ ;

$F$ : conjunto de estados finais  $F \subseteq Q$ .

onde:

$\delta$  é a função de transição que estabelece a maneira como deve ser feita a mudança de estado, sendo  $q \in Q$  o estado atual de  $M$  e  $\sigma \in \Sigma$  o símbolo corrente na fita de entrada, então  $\delta(q, \sigma) \in Q$  determina um único estado para o qual  $M$  irá se mover.

$+_M$  é uma relação binária entre configurações. Estabelece todas as situações possíveis de mudança de configuração interna em um só movimento. Isto é,  $+_M$  representa as transições de estado permitidas pelo controle finito.

O movimento  $(q, w) +_M (q', w')$  ocorre se e somente se  $w = \sigma w'$  e  $\delta(q, \sigma) = q'$ , para  $\sigma \in \Sigma$ , ou seja, o movimento de um estado para o seguinte, consumindo um símbolo da cadeia de entrada, ocorre quando este símbolo pertence ao alfabeto de entrada e a função  $\delta$  determina qual será o próximo estado.

$+_M^*$  denota o fechamento recursivo e transitivo de  $+_M$ , isto é, podemos operar  $+_M$  quantas vezes for necessário para o consumo da cadeia de entrada. Esta seqüência é chamada de uma

computação executada por  $M$  sobre  $w$ , e tem comprimento  $n$ , que representa o número de passos necessários para o consumo da cadeia.

A cadeia  $w \in \Sigma^*$  será aceita por  $M$  se e somente se existir um  $q \in F$ , tal que  $(q_0, w) \vdash_M^* (q, \epsilon)$ .

Por último, a linguagem aceita pelo autômato  $M$  é o conjunto de todas as cadeias aceitas por  $M$  e é denotada como  $L(M)$ .

## Anexo B

### Descrição formal dos autômatos finitos não-determinísticos

Seja o autômato finito não-determinístico  $M = \{Q, \Sigma, \Delta, q_0, F\}$

com:

$Q$ : conjunto finito de estados;

$\Sigma$ : alfabeto de entrada;

$\Delta$ : relação de transições  $\Delta \subseteq Q \times \Sigma^* \times Q$ ;

$q_0$ : estado inicial único  $q_0 \in Q$ ;

$F$ : conjunto de estados finais  $F \subseteq Q$ .

onde:

$\Delta$  é a relação de transições, sendo subconjunto finito de  $Q \times \Sigma^* \times Q$ .

$(q, u, p) \in \Delta$  é chamada de transição de  $M$ , pois estando no estado  $q$ ,  $M$  consome a cadeia de entrada  $u$  (que pode ou não ser composta de apenas um símbolo) e transita para o estado  $p$ .

$+_M$  é definida como  $(q, w) +_M (q', w') \Leftrightarrow \exists u \in \Sigma^* \mid w = u \circ w' \wedge (q, u, q') \in \Delta$ . Isto é, a transição do autômato  $M$  ocorre de  $q$  para  $q'$  se e somente se existir uma cadeia  $u$  na fita de entrada e  $\Delta$  contenham a relação  $(q, u, q')$ .

A cadeia  $w \in \Sigma^*$  será uma aceita por  $M$  se e somente se existir  $q \in F$ , tal que  $(q_0, w) +_M^* (q, \epsilon)$ .

A cadeia é aceita pelo autômato finito não-determinístico quando após todos os movimentos necessários do controle finito, o último símbolo da cadeia seja consumido e o estado final do autômato seja um estado válido de aceitação.

Por último, a linguagem aceita pelo autômato  $M$  é o conjunto de todas as cadeias aceitas por  $M$  e é escrita como  $L(M)$ .

Cabe observar que os autômatos finitos determinísticos são casos particulares dos autômatos finitos não-determinísticos se na relação  $(q,u,q') \in \Delta$  tivermos:

- $|u| = 1$
- para  $q \in Q$  e  $\sigma \in \Sigma$ , houver um único  $q' \in Q$  tal que  $(q,\sigma,q') \in \Delta$ .

## Anexo C

### Descrição formal dos autômatos finitos de pilha

Define-se autômatos de pilha como uma sêxtupla:  $M = \{Q, \Sigma, \Gamma, \Delta, q_0, F\}$

com:

$Q$ : conjunto finito de estados;

$\Sigma$ : alfabeto de entrada;

$\Gamma$ : alfabeto de pilha;

$\Delta$ : relação de transição de estados  $\Delta \subset (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$ ;

$q_0$ : estado inicial único  $q_0 \in Q$ ;

$F$ : conjunto de estados finais  $F \subseteq Q$ .

onde:

$\Delta$  é a relação não-determinística de transição que dita como será a mudança de estado e o que será adicionado na pilha, a partir do estado atual, do símbolo de entrada e do símbolo corrente no topo da pilha: se o par ordenado  $((p, u, \beta), (q, \gamma)) \in \Delta$ , então estando o autômato  $M$  no estado  $p$ , como  $\beta$  no topo da pilha, poderá consumir  $u$  da cadeia de entrada e evoluir para o estado  $q$ , modificando para  $\gamma$  o topo da pilha.

$M$  aceita  $w \in \Sigma^*$  se e somente se  $(q_0, w, \epsilon) +_M^* (p, \epsilon, \epsilon)$ , com  $p \in F$ .

$+_M^*$  representa o fechamento recursivo e transitivo de  $+_M$ .

A linguagem aceita pelo autômato  $M$  é o conjunto de todas as cadeias aceitas por  $M$  e é escrita como  $L(M)$ , com:

$L(M) = \{w \in \Sigma^* \mid (q_0, w, \epsilon) +_M^* (p, \epsilon, \epsilon), p \in F\}$ .

## SUMÁRIO

1.	INTRODUÇÃO .....	1
1.1	Objetivo .....	2
1.2	Justificativas.....	2
1.3	Conteúdo e Organização.....	5
2.	PESQUISAS NA ÁREA DE MAPEAMENTO E EXPLORAÇÃO DE AMBIENTES ..	6
2.1	Utilização de autômatos na robótica móvel .....	6
2.2	Mapeamento de ambientes .....	7
2.3	Exploração de ambientes desconhecidos.....	12
2.4	Cobertura completa de ambientes por robôs móveis.....	14
3.	O MODELO PROPOSTO.....	17
4.	AUTÔMATOS ADAPTATIVOS.....	19
4.1	Introdução .....	19
4.2	Autômatos Finitos.....	20
4.2.1	Autômatos finitos determinísticos .....	20
4.2.2	Autômatos finitos não-determinísticos .....	21
4.2.3	Propriedades das linguagens aceitas por autômatos finitos .....	22
4.3	Autômatos de Pilha e Linguagens Livres de Contexto.....	22
4.3.1	Linguagens livres de contexto .....	22
4.3.2	Autômatos de pilha.....	22
4.4	Autômatos Adaptativos e Linguagens Sensíveis ao Contexto .....	23
4.4.1	Linguagens sensíveis ao contexto.....	23
4.4.2	Autômatos Adaptativos.....	23
5.	AUTÔMATO DE MAPEAMENTO.....	29
5.1	Introdução .....	29
5.2	Configuração Inicial do Autômato de Mapeamento .....	32
5.3	Operação do Autômato .....	33
5.4	Descrição do Autômato de Mapeamento.....	37
5.5	Considerações sobre o Autômato de Mapeamento .....	53
6.	AUTÔMATO DE EXPLORAÇÃO.....	55
6.1	Introdução .....	55
6.2	Comportamento do Autômato de Exploração .....	59
6.2.1	Comportamento do Autômato de Exploração utilizando marcos.....	62
6.3	Configuração do Autômato de Exploração .....	65
6.4	Considerações sobre o Autômato de Exploração .....	88

7.	SIMULAÇÕES.....	91
7.1	Primeira simulação: ambiente com características simples .....	91
7.2	Segunda simulação: ambiente típico .....	94
7.3	Terceira simulação: ambiente atípico .....	97
7.4	Quarta simulação: ambiente-teste de mínimo local .....	101
7.5	Quinta simulação: ambiente-teste de deslocamento contínuo .....	104
7.6	Considerações .....	106
8.	CONCLUSÕES .....	115
9.	TRABALHOS FUTUROS .....	117
9.1	Navegação .....	117
9.2	Outras perspectivas .....	119
10.	REFERÊNCIAS BIBLIOGRÁFICAS .....	124
	Apêndice A .....	134
	Apêndice B .....	140
	Anexo A.....	153
	Anexo B.....	155
	Anexo C .....	157