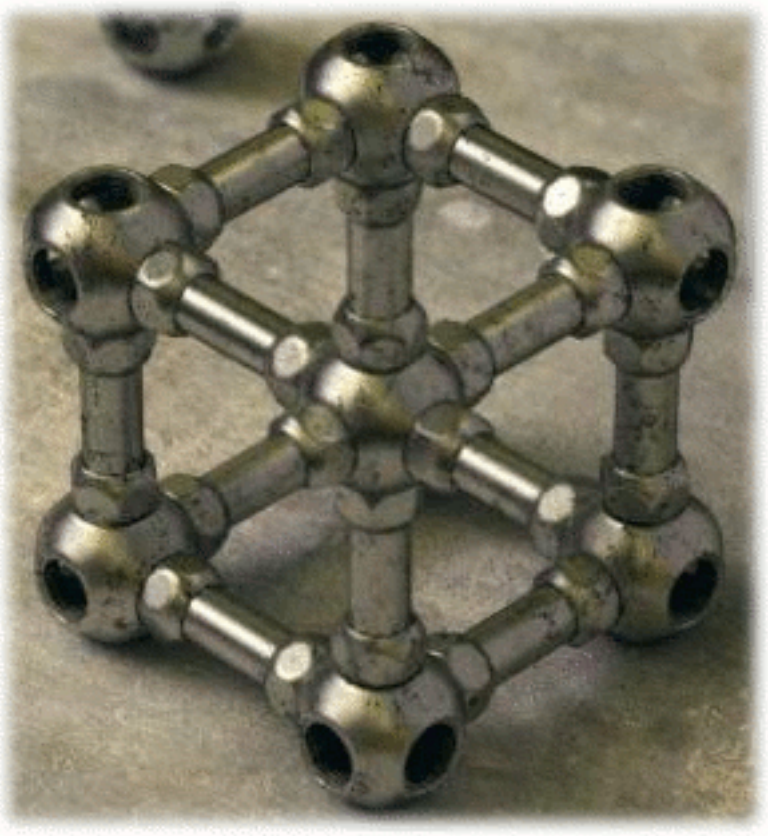

# Aprendizado por Reforço Relacional para Reaproveitamento do Conhecimento em Navegação Robótica [1]

**Author(s):**
Tiago Matos
Anna Helena Reali Costa

[1]This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

# Aprendizado por Reforço Relacional para Reaproveitamento do Conhecimento em Navegação Robótica

Tiago Matos[1] e Anna Helena Reali Costa[1]

Laboratório de Técnicas Inteligentes LTI, Escola Politécnica da Universidade de São Paulo EPUSP

**Resumo** Soluções propostas para o aprendizado na navegação robótica geralmente não levam em conta soluções previamente obtidas em tarefas similares ao problema que se quer resolver. As que reutilizam as soluções previamente encontradas tentam aproveitar o conhecimento através de abstração dos estados que possuem o mesmo valor na função utilidade considerada na tarefa, não aproveitando para isso as relações existentes entre os objetos do ambiente. Este trabalho visa utilizar o aprendizado por reforço relacional, e as técnicas de abstração propostas para ele, no reaproveitamento do conhecimento aprendido na solução do problema, visando com isso obter uma política abstrata para o problema de navegação que possa ser utilizada diretamente em tarefas similares.

**Keywords:** Aprendizado Relacional, Reaproveitamento de políticas, Navegação Autônoma, Robótica

## 1 Introdução

Processos Markovianos de Decisão (MDP, do inglês *Markov Decision Processes*) podem ser utilizados para modelar a dinâmica e solucionar problemas importantes do mundo real, nos quais as ações tomadas são estocásticas, como a navegação de um robô por um ambiente, onde a ação de mover de uma sala para outra pode ou não ter sucesso. MDPs são utilizados para encontrar uma política, ou seja, apresentar uma ação para um determinado estado em que se encontra o sistema a ser controlado. Embora útil na modelagem de vários problemas, os MDPs sofrem do mal da dimensionalidade, já que a quantidade de estados cresce exponencialmente em função da quantidade de atributos utilizados na representação destes. Esse mal ocorre tanto quando se tem o modelo completo do processo quanto quando não se tem o modelo. Ainda, uma característica de uma política encontrada como solução para um problema é que a mesma não pode ser diretamente reaproveitada para a resolução de problemas similares.

Em vista disto, diversas soluções têm sido propostas para a generalização e reaproveitamento do conhecimento visando contornar algum ou ambos destes problemas. Essas técnicas podem ser divididas em duas linhas de pesquisa. A primeira realiza a generalização do conhecimento através da criação de macro-estados e(ou) macro-ações, como o trabalho apresentado por Drummond [2002]

que realiza a agregação de estados que se encontram numa determinada região da função valor obtida através do aprendizado por reforço e divide a superfície desta em macro-ações, para posterior utilização das macro-ações na resolução de tarefas semelhantes, ou o trabalho de Uther e Veloso [2002] no qual políticas para sub-tarefas de um determinado problema são utilizadas para a criação de ações abstratas e estas são utilizadas conjuntamente com uma árvore que representa a agregação de estados do sistema. A segunda realiza a generalização do conhecimento através da geração de estados abstratos e(ou) ações abstratas, feita através de uma representação relacional dos estados e ações do problema. Nessa segunda linha se encontram os trabalhos de: Dzeroski, Raedt e Driessens [2001] e Kersting et al. [2007]. Nesses trabalhos, os estados abstratos são representados em uma árvore de decisão em lógica de primeira ordem, pela conjunção dos predicados nos nós internos. As folhas da árvore representam as ações abstratas. Outro trabalho nessa segunda linha, é o de Driessens e Ramon [2003], no qual a agregação de estados é feita através da utilização de métodos de Aprendizado Baseado em Instâncias (IBL - *Instance Based Learning*) [Aha, Kibler e Albert 1991].

Aprendizado por reforço relacional (ARR) [Dzeroski, Raedt e Driessens 2001] é uma técnica que combina o aprendizado por reforço [Sutton e Barto 1998] e a programação lógica indutiva. Assim como o aprendizado por reforço, essa técnica foi proposta para a resolução de um problema quando não se tem o modelo completo do processo. O objetivo do ARR é aprender uma política abstrata, ou seja, uma política para a escolha de uma ação abstrata ótima para um determinado estado abstrato. O interesse aqui consiste no uso do aprendizado por reforço relacional no domínio da robótica móvel. Algumas propostas já aplicaram o aprendizado por reforço relacional no problema do mundo dos blocos visando abstrair os estados em função das relações entre os blocos no ambiente. Apesar de existirem propostas para o reaproveitamento do aprendizado no domínio da robótica móvel [Drummond 2002] essas propostas não são baseadas na agregação ou reaproveitamento do conhecimento decorrente das relações existentes entre os objetos do ambiente. Isso limita a aplicação dessas técnicas apenas à navegação pura no ambiente, não podendo adicionar tarefas mais complexas na tarefa do robô, como a entrega de objetos para determinadas pessoas, ou a possibilidade de ações mais complexas. Além disso não possibilita a utilização direta do conhecimento aprendido num ambiente com características semelhantes. A única utilização conhecida de uma técnica de abstração relacional no problema de navegação robótica foi o trabalho de Kersting et al. [2007]. Nesse trabalho uma política abstrata era induzida através de pares de estado/ação ótimos obtidos pela resolução do MDP que representava um determinado problema de navegação. Em vista disto, o objetivo deste trabalho é utilizar o aprendizado por reforço relacional, e duas técnicas de abstração propostas para gerar uma política abstrata que consiga representar as relações existentes no ambiente para uma determinada tarefa de navegação no domínio da robótica móvel. Pretende-se utilizar métodos não-incrementais de indução dos algoritmos de regressão, pois é considerado no trabalho que a solução (política) para um determinado problema já foi aprendida e deseja-se reutilizar esse conhecimento, através da abstração da política,

na solução de tarefas similares. Em vista deste fato, pretende-se utilizar para a abstração dos dados os algoritmos TILDE [Dzeroski, Raedt e Driessens 2001] e RIB' (uma versão não incremental do algoritmo RIB [Driessens e Ramon 2003] proposta neste trabalho).

Na atual fase do trabalho, dois algoritmos de regressão propostos para o aprendizado por reforço relacional, o TILDE e o RIB, foram aplicados no domínio do mundo dos blocos. Pretende-se, nas próximas etapas do projeto, aplicar os algoritmos de regressão num problema de navegação robótica para analisar como estes sistemas abstraem os diversos estados do problema e, com isso, propor uma solução adequada para a política instanciada não-ótima que pode surgir quando uma abstração relacional é utilizada num problema de navegação robótica.

O restante do artigo é organizado da seguinte forma. A seção 2 descreve o Processo Markoviano de Decisão Relacional. A seção 3 descreve a abstração relacional para um MDP Relacional. A seção 4 apresenta resultados preliminares obtidos no Mundo dos Blocos. A seção 5 apresenta a modelagem relacional para um problema de navegação. E finalmente a seção 6 apresenta os próximos passos do projeto.

## 2 Processo Markoviano de Decisão Relacional

A representação relacional (ou estrutural) de um problema tem como objetivo representar as características mais importantes dos objetos e das relações existentes entres estes no ambiente (ou mundo) sobre o qual o problema está sendo considerado [Driessens 2005]. Essa difere em diversos pontos de outras representações utilizadas, como a representação por um vetor de atributos, para modelar um problema. A representação de um estado não está limitada pela quantidade de atributos do vetor, pois a representação é feita por uma conjunção de predicados. Por utilizar uma conjunção de predicados, essa representação possui como vantagem, em relação a outras, a fácil visualização da definição de um estado, mas possui a desvantagem de utilizar uma quantidade maior de memória nessa representação. Num problema que está sendo representado por um MDP, tanto os estados quanto as ações podem ser representados em sua forma relacional. Por exemplo, a Figura 1 apresenta um estado e uma ação possível para este estado representados na forma relacional.

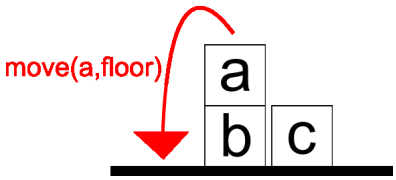


**Figura 1.** Representação relacional de um estado no Mundo dos Blocos, dada por **on(a,b)** $\wedge$ **clear(a)** $\wedge$ **clear(c)** $\wedge$ **on(b,floor)** $\wedge$ **on(c,floor)** e da ação de mover o bloco a para o chão, dada pelo predicado **move(a,floor)**

Um MDP Relacional [Kersting, Otterlo e Raedt 2004] é definido formalmente por uma quádrupla $\langle S,\ A,\ T,\ R\rangle$ onde:

- $S$ é um conjunto de estados abstratos representados na forma relacional;
- $A$ é um conjunto de ações abstratas representadas na forma relacional;
- $T$ é uma função de transição, T: $S \times A \rightarrow S$;
- $R$ é uma função de recompensa abstrata, R: $S \rightarrow \mathbb{R}$;

Um estado abstrato é uma conjunção de predicados onde os termos destes são variáveis. Ele tem a função de representar um conjunto de estados instanciados (estados representados na forma proposicional). Por exemplo, no Mundo dos Blocos, o estado abstrato **clear(X)** $\wedge$ **clear(Y)** representaria todos os estados que possuem dois blocos livres para movimentação. Uma ação abstrata é um conjunto finito de regras de ações $H_i \overset{p_i:C}{\leftarrow} B$ no qual C é um literal que representa o nome e os argumentos da ação, B é um estado abstrato contendo as pré-condições da ação C, $H_i$ é o i-ésimo efeito possível da ação, para a aplicação da ação C, e $p_i$ é a probabilidade do efeito da ação ser $H_i$. Por exemplo, considerando novamente o estado abstrato **clear(X)** $\wedge$ **clear(Y)** uma possível ação abstrata para este estado seria:

$clear(X) \wedge on(X,Y) \wedge notclear(Y) \overset{1:move(X,Y)}{\longleftarrow} clear(X) \wedge clear(Y)$

A função de recompensa abstrata especifica a recompensa gerada pela entrada em estados abstratos. Um exemplo de uma recompensa abstrata R(s), para o estado s: $clear(X) \wedge on(X,Y)$, seria:

**R(s):** $1 \leftarrow clear(X) \wedge on(X,Y)$

A seção a seguir apresentará dois algoritmos para abstração de estados que utilizam a representação relacional aqui definida.

## 3 Abstração Relacional de Políticas

Os algoritmos de regressão citados nesta seção foram inicialmente propostos para serem utilizados conjuntamente com o Aprendizado por Reforço Relacional ARR [Dzeroski, Raedt e Driessens 2001]. O TILDE que é um algoritmo não incremental de indução de árvore de decisão em linguagem de primeira ordem. Este algoritmo foi escolhido no trabalho por ser um método não incremental. O RIB é um algoritmo de indução baseado em exemplos e foi escolhido devido à sua simplicidade [Driessens 2005]. Nos itens 3.1 e 3.2 a seguir são explicados estes algoritmos.

### 3.1 Abstração em Árvore de Decisão

Árvore de Decisão é uma das estruturas mais populares para a representação do conhecimento e dos padrões estruturais encontrados em dados obtidos pelas técnicas de Aprendizagem de Máquina [Mitchell 1997,Witten e Frank 2005]. Nesse tipo de estrutura os nós internos têm a função de testar um determinado atributo e as folhas representam as classes (para a classificação dos dados) ou

valores reais (para a regressão dos dados) para todos os exemplos que são classificados nestas folhas. As árvores de decisão em lógica de primeira ordem são uma adaptação das árvores de decisão proposicionais para a lógica de primeira ordem [Driessens 2005], onde as seguintes alterações são feitas na estrutura da árvore:

- Os nós internos da árvore possuem, cada um, uma conjunção de literais de primeira ordem.
- Os nós podem compartilhar variáveis, tendo a restrição de que uma variável que foi introduzida (criada) num determinado nó só pode aparecer na subárvore esquerda, ou seja, aparece apenas na subárvore onde o teste do nó foi válido.

A figura 2 exemplifica uma árvore de decisão em lógica de primeira ordem. As folhas desta árvore (V1,V2 e V3) contêm a classe representativa (classificação) ou o valor médio (regressão) dos exemplos que são classificados nesta folha.

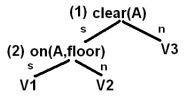


**Figura 2.** Árvore de Decisão em Lógica de Primeira Ordem

O TILDE [Blockeel e Raedt 1998] é um algoritmo não incremental de indução de árvores de decisão que é uma extensão para a lógica de primeira ordem do algoritmo TDIDT (*Top-Down Induction of Decision Trees*) [Quinlan 1986]. Dado um conjunto de exemplos a serem divididos, o TILDE gera um conjunto de testes candidatos para a divisão dos exemplos. Os testes são criados a partir de um operador de refinamento. Após isso, é selecionado o melhor teste entre os candidatos. A seleção do melhor teste é feita através do *default gain ratio* [Quinlan 1993]. Depois da seleção do teste, é criado um nó interno na árvore e dois subconjuntos (subárvores desse nó): o primeiro contendo os exemplos no qual o teste é válido e o segundo contendo os exemplos onde o teste é inválido. Por fim, as etapas descritas anteriormente são repetidas para cada um dos subconjuntos criados. Essas etapas se repetem até o critério de parada da indução do algoritmo ser válido. No caso do TILDE, quando não existe um conjunto mínimo de exemplos em cada um dos subconjuntos ou quando nenhum teste apresentar ganho de informação significativo. Para cada folha da árvore é estimado um valor (ou uma classe) que represente os exemplos existentes nesse conjunto.

### 3.2 Abstração em aprendizado baseado em exemplos

O método de aprendizado baseado em exemplos (*Instance Based Learning* IBL) é um método de aprendizado que apenas armazena os exemplos que são passados durante o seu treinamento. Cada vez que um exemplo deve ser classificado (ou

estimado o seu valor, se for uma regressão), ele é comparado com todos os outros armazenados pelo sistema.

O algoritmo RIB [Driessens e Ramon 2003] é um método IBL que foi proposto para ser utilizado no aprendizado por reforço relacional. Assim como os outros sistemas IBL, ele funciona basicamente armazenando exemplos e comparando estes para apresentar uma estimativa $\hat{Q}(s,a)$ para um valor $Q(s,a)$ para um determinado par estado/ação. Os exemplos passados para o RIB são representados relacionalmente. Em vista deste fato, para o cálculo da distância entre exemplos é proposta uma medida de distância relacional entre estes. O cálculo de distância relacional entre os exemplos tem como objetivo o reaproveitamento do conhecimento prévio existente sobre a estrutura e a dinâmica do processo, sendo este cálculo dependente do domínio no qual o algoritmo RIB vai ser aplicado. Uma proposta de distância relacional para o domínio do Mundo dos Blocos é apresentada em Driessens e Ramon [2003].

No aprendizado por reforço relacional, o RIB é utilizado conjuntamente com a representação relacional para abstrair os pares de estado/ação armazenados em uma política abstrata. São armazenados apenas os exemplos que possuem valor Q(s,a) distante do valor que é estimado pelo RIB. A estimativa para o valor Q(s,a) de um determinado exemplo é dada pela média ponderada, em função da distância relacional, de todos os valores dos exemplos armazenados, isto é, $\hat{q_i} = \frac{\sum_j \frac{q_j}{dist_{ij}}}{\sum_j \frac{1}{dist_{ij}}}$, sendo $q_j$ o valor Q(s,a) do exemplo armazenado e $dist_{ij}$ a distância relacional entre o par estado/ação que se quer estimar o valor Q(s,a) e o exemplo armazenado pelo RIB.

## 4 Estudo no Mundo dos Blocos

Os algoritmos de regressão foram utilizados no problema do Mundo dos Blocos com o intuito de verificar como os pares de estado/ação que são passados na indução são abstraídos por estes. Inicialmente, um algoritmo de aprendizado por reforço Q-learning foi utilizado para o treinamento e convergência dos valores de uma tabela Q em um problema com a meta de colocar um determinado bloco intitulado $a$ em cima de outro $b$ para um problema contendo 4 blocos. Após a convergência da tabela, esta foi utilizada em 1000 episódios de validação (onde, para cada estado, era escolhida a ação ótima a ser aplicada neste). Cada par de estado/ação percorrido em cada um destes episódios foi armazenado. A partir dessas informações foram realizados dois tipos de testes. No primeiro teste, todo o conjunto de pares visitados durante os episódios de validação foi passado para o sistema de regressão TILDE (semelhante aos testes realizados em [Kersting et al. 2007]) e, no segundo, todos os pares de estado/ação da tabela Q foram passados para o sistema RIB.

Os quatro estados abstratos gerados pelo TILDE no primeiro teste estão indicados na Figura 3, onde os blocos de cor preta representam um conjunto de um ou mais blocos[1].

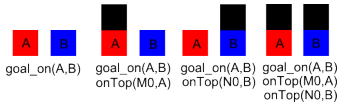


**Figura 3.** Estados abstratos gerados pela indução do TILDE

A Figura 4.a representa a função valor, codificada em tonalidade de cinza, para o problema com meta de colocar o bloco $a$ em cima do $b$. Após o aprendizado, os valores Q(s,a) para todos os pares de estado/ação existentes no problema foram plotados numa imagem (Figura 4.a). Nessa imagem, cada quadrado corresponde a um par estado/ação para o problema. Cada linha dessa imagem corresponde a um estado possível. Cada coluna corresponde a uma ação possível, por exemplo, a primeira coluna é a ação $move(a, floor)$, a segunda coluna é a ação $move(a, b)$ ... , a quinta coluna é a ação $move(b, floor)$, a sexta coluna é a ação $move(b, a)$, e assim sucessivamente. A escala de cinza indica a faixa de valores para Q, indo de Q igual a 1 (branco) até 0 (preto).

A Figura 4.b mostra como os estados instanciados foram agregados pelos estados abstratos gerados pelo algoritmo TILDE. O TILDE abstraiu o conjunto de estados do problema em seus diversos estados abstratos. Os estados abstratos foram: **goal_on(A,B)**, **goal_on(A,B)** $\wedge$ **onTop(MO,A)**, **goal_on(A,B)** $\wedge$ **onTop(NO,B)**, **goal_on(A,B)** $\wedge$ **onTop(M0,A)** $\wedge$ **onTop(N0,B)**. Estes estados abstratos estão respectivamente indicados na Figura 4.b pelas cores branco, cinza claro, cinza médio e cinza escuro. Eles podem ser classificados em quatro distâncias (situações) em relação à meta de colocar um determinado bloco A em cima de outro B: a mais próxima da meta onde ambos os blocos estão livres e é necessária apenas uma ação pra atingir a meta, a que o bloco A está obstruído e deve-se remover os blocos que estão em cima deste, a que o bloco B está obstruído e deve-se remover os blocos que estão em cima deste, e finalmente a que ambos os blocos estão obstruídos.

Os 50 exemplos mantidos pelo RIB estão indicados na Figura 4.c, ou seja, do total de pares de estado/ação presentes na tabela Q, apenas os indicados (brancos) foram armazenados pelo RIB. Não há um padrão específico na função valor, da tabela Q, que indique quais desses exemplos serão armazenados. A abstração de um estado pelo RIB é feita durante a etapa de renomeação. Em vista disto, diferentes pares de estado/ação com valores distantes para Q(s,a) podem ser classificados pelo mesmo estado abstrato. Devido a esse fato, e pelo critério de

[1] **Predicado goal_on(X,Y)**: indica que a meta para o problema é colocar o bloco X sobre o bloco Y. **Predicado on(X,Y)**: indica que o bloco X está sobre o bloco Y. **Predicado onTop(X,Y)**: indica que os blocos X e Y estão na mesma pilha de blocos e X está no topo desta.

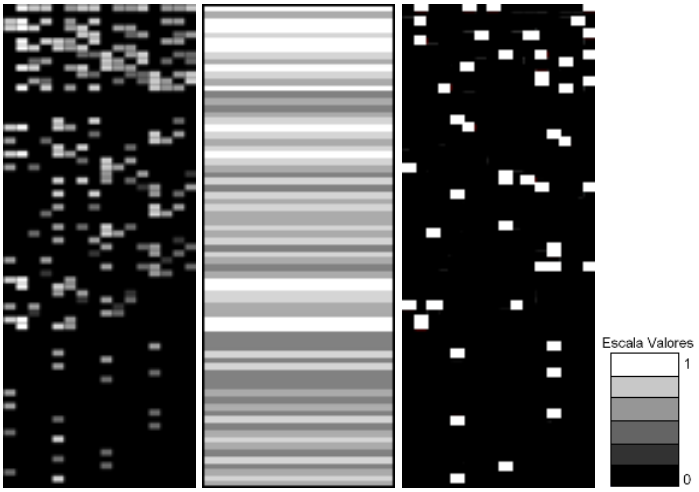


**Figura 4. (a)** Função Valor Q(s,a) para o problema do mundo dos blocos contendo 4 blocos. **(b)** Agregação de estados pelo algoritmo TILDE. Divisão dos estados instanciados pelos estados abstratos. **(c)** Agregação de estados pelo algoritmo RIB. Exemplos armazenados.

armazenamento ser baseado no desvio-padrão dos valores Q(s,a), exemplos com diferentes valores de Q(s,a) representados pelo mesmo estado abstrato podem ser armazenados pelo RIB. Um exemplo desta situação são os pares estado/ação **on(b,floor)** $\wedge$ **on(c,floor)** $\wedge$ **on(d,floor)** $\wedge$ **on(a,d)** $\wedge$ **clear(b)** $\wedge$ **clear(c)** $\wedge$ **clear(a) / move(b,a)** e **on(a,floor)** $\wedge$ **on(c,floor)** $\wedge$ **on(d,floor)** $\wedge$ **on(b,d)** $\wedge$ **clear(a)** $\wedge$ **clear(c)** $\wedge$ **clear(b) / move(a,b)** que possuem distintos valores para Q(s,a), mas são representados pelo mesmo estado abstrato. Devido à indistinção nos exemplos, pode ser gerada uma política não-ótima para o problema que se quer resolver. Um exemplo desse problema ocorreu para o estado **on(a,c)** $\wedge$ **clear(a)** $\wedge$ **clear(b)** $\wedge$ **clear(d)**. Para esse estado o valor Q(s,a) fornecido pelos exemplos do RIB para a ação $move(b, a)$ foi maior do que o fornecido para qualquer uma das outras ações possíveis neste estado. A ação $move(b, a)$ forneceu um valor $Q(s, a)$ igual a 0.904 enquanto que a ação $move(a, b)$, que seria ótima para este estado, forneceu um valor para $Q(s, a)$ igual a 0.902.

Conforme resultados anteriores [Kersting et al. 2007,Driessens 2005] a abstração relacional realizada pelo TILDE gerou a política ótima para o problema de colocar um determinado bloco em cima de outro no mundo dos blocos, para qualquer quantidade de blocos existentes no problema, enquanto que a abstração realizada pelo RIB não apresenta um desempenho ótimo para esta tarefa. O interesse dos testes realizados no domínio do Mundo dos Blocos foi basicamente o estudo e análise desses algoritmos. Através desses testes foi possível se inteirar melhor dos algoritmos e do funcionamento destes, avaliando, em um domínio

mais simples, as agregações e abstrações feitas. A seguir será descrita a modelagem relacional para o problema de interesse do trabalho.

## 5 Modelagem Relacional do Problema de Navegação Robótica

A modelagem relacional para um problema de navegação robótica que o presente trabalho utiliza é a proposta em Kersting et al. [2007]. Nesse trabalho, o domínio de navegação é o Mundo do Hotel. Nesse problema o agente se encontra num hotel e tem como meta chegar numa determinada localidade, como por exemplo a entrada deste. Os locais podem ser de três tipos: **room(R)**, **vertical_Passage(V)** e **horizontal_Passage(H)**[2]. A definição de um estado para esse problema é dada pela localidade que o agente se encontra e pelas localidades conectadas à esta[3]. As ações possíveis para o problema são todas as combinações de movimentação entre as localidades. Como exemplo, a movimentação de uma **room**($r_1$) para uma **vertical_Passage**($v_1$) seria dada pelo predicado **goto_RVP**($r_1, v_1$), a movimentação de uma **vertical_Passage**($v_3$) para uma **room**($r_2$) por **goto_VPR**($v_3, r_2$).

Através desta representação, os pares de estado/ação podem ser utilizados por um algoritmo de regressão para apresentar uma política abstrata para o problema de navegação robótica.

## 6 Próximos Passos

Pretende-se aplicar e comparar o desempenho destes dois algoritmos de regressão no domínio da navegação robótica. O domínio da navegação robótica apresenta características próprias que torna desafiadora a aplicação de um algoritmo de regressão relacional para o aprendizado de uma política abstrata. A relação existente entre os objetos numa representação relacional para o domínio de navegação robótica pode levar a uma política abstrata que, quando é instanciada, torna-se não-determinística. Uma aplicação do TILDE para o aprendizado de uma política abstrata de navegação foi feita em [Kersting et al. 2007]. Para essa política abstrata aprendida, estando numa determinada sala e sendo a ação abstrata ótima a ser aplicada "ir para uma passagem horizontal", a instanciação dessa política abstrata ótima para a resolução de um determinado problema não consegue diferenciar qual dessas passagens horizontais deveria ser escolhido pela política ótima, devido à perda de informação por causa da abstração realizada e no caso de ter mais de uma passagem conectada à sala em que o robô se encontra. Entretanto, no domínio de navegação robótica, a escolha de um caminho qualquer entre as opções poderia levar a uma política instanciada não-ótima.

[2] **Predicado room(R)**: o agente se encontram na sala R. **Predicado vertical_Passage(V)**: o agente se encontra na passagem vertical V. **Predicado horizontal_Passage(H)**: o agente se encontra na passagem horizontal H

[3] **Predicado connected**($R_1, R_2$): indica que a localidade $R_1$ e $R_2$ estão conectadas

Pretende-se no desenvolvimento deste trabalho de pesquisa, propor modificações, seja na representação relacional do problema de navegação ou nos algoritmos de regressão, onde isso possa ser considerado. Espera-se utilizar a política abstrata aprendida em problemas similares, tais como quando há mudança do local alvo em um mesmo ambiente.

## 7 Agradecimentos

Os autores agradecem ao CNPq Proc. N. 305512/2008-0 e à FAPESP Proc. N. 2008/03995-5 e Proc. N. 2009/04489-9 pelo apoio recebido.

## Referências

[Aha, Kibler e Albert 1991]AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. In: *Machine Learning*. [S.l.: s.n.], 1991. p. 37–66.

[Blockeel e Raedt 1998]BLOCKEEL, H.; RAEDT, L. D. Top-down induction of logical decision trees. *Artificial Intelligence*, v. 101, p. 285–297, May 1998. ISSN 0004-3702.

[Driessens 2005]DRIESSENS, K. Thesis: relational reinforcement learning. *AI Communications*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 18, p. 71–73, January 2005. ISSN 0921-7126.

[Driessens e Ramon 2003]DRIESSENS, K.; RAMON, J. Relational instance based regression for relational reinforcement learning. In: *In Proceedings of the 20th International Conference on Machine Learning*. [S.l.]: AAAI Press, 2003. p. 123–130.

[Drummond 2002]DRUMMOND, C. Accelerating reinforcement learning by composing solutions of automatically identified subtasks. *Journal of Artificial Intelligence Research*, v. 16, p. 59–104, February 2002. ISSN 1076-9757.

[Dzeroski, Raedt e Driessens 2001]DZEROSKI, S.; RAEDT, L. D.; DRIESSENS, K. Relational reinforcement learning. *Machine Learning*, v. 43, p. 7–52, 2001.

[Kersting, Otterlo e Raedt 2004]KERSTING, K.; OTTERLO, M. V.; RAEDT, L. D. Bellman goes relational. In: *Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM, 2004. (ICML '04), p. 465–472.

[Kersting et al. 2007]KERSTING, K. et al. Learning to transfer optimal navigation policies. *Advanced Robotics: Special Issue on Imitative Robots*, v. 21, n. 13, p. 1565—1582, 2007.

[Mitchell 1997]MITCHELL, T. M. *Machine Learning*. 1. ed. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. Hardcover. ISBN 0070428077.

[Quinlan 1986]QUINLAN, J. R. Induction of decision trees. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 81–106, 1986. ISSN 0885-6125.

[Quinlan 1993]QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

[Sutton e Barto 1998]SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998. ISBN 0262193981.

[Uther e Veloso 2002]UTHER, W. T. B.; VELOSO, M. M. Ttree: Tree-based state generalization with temporally abstract actions. In: *In Proceedings of SARA-2002*. [S.l.: s.n.], 2002. p. 260–290.

[Witten e Frank 2005]WITTEN, I. H.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. 2. ed. [S.l.]: Morgan Kaufmann, 2005. Paperback. ISBN 0120884070.