TOMLG

Lucas S. da Simões Matheus F. Ferraz
Orientador: Gustavo A. Giménez Lugo

UTFPR

December 3, 2018

Sumário

Pergunta

Como encarar um problema cognitivo de teoria da mente como um problema computacional?

Objetivos

Objetivo Geral:

- Propor um modelo computacional que incorpore aprendizagem e raciocínio baseado no modelo lógico de ToM de Lorini (2007)

Objetivos Específicos:

- Instanciar uma estrutura baseada em reinforcement learning passível de ser incorporada a uma base de conhecimento lógico;
- Extensão em uma lógica que permite o raciocínio perceptual como parte de suas operações;
- Verificação das condições de aprendizagem e raciocínio em um cenário perceptual.

Theory of mind

- O que é Theory of Mind?
- Experimento Sally e Anne.

Theory of mind

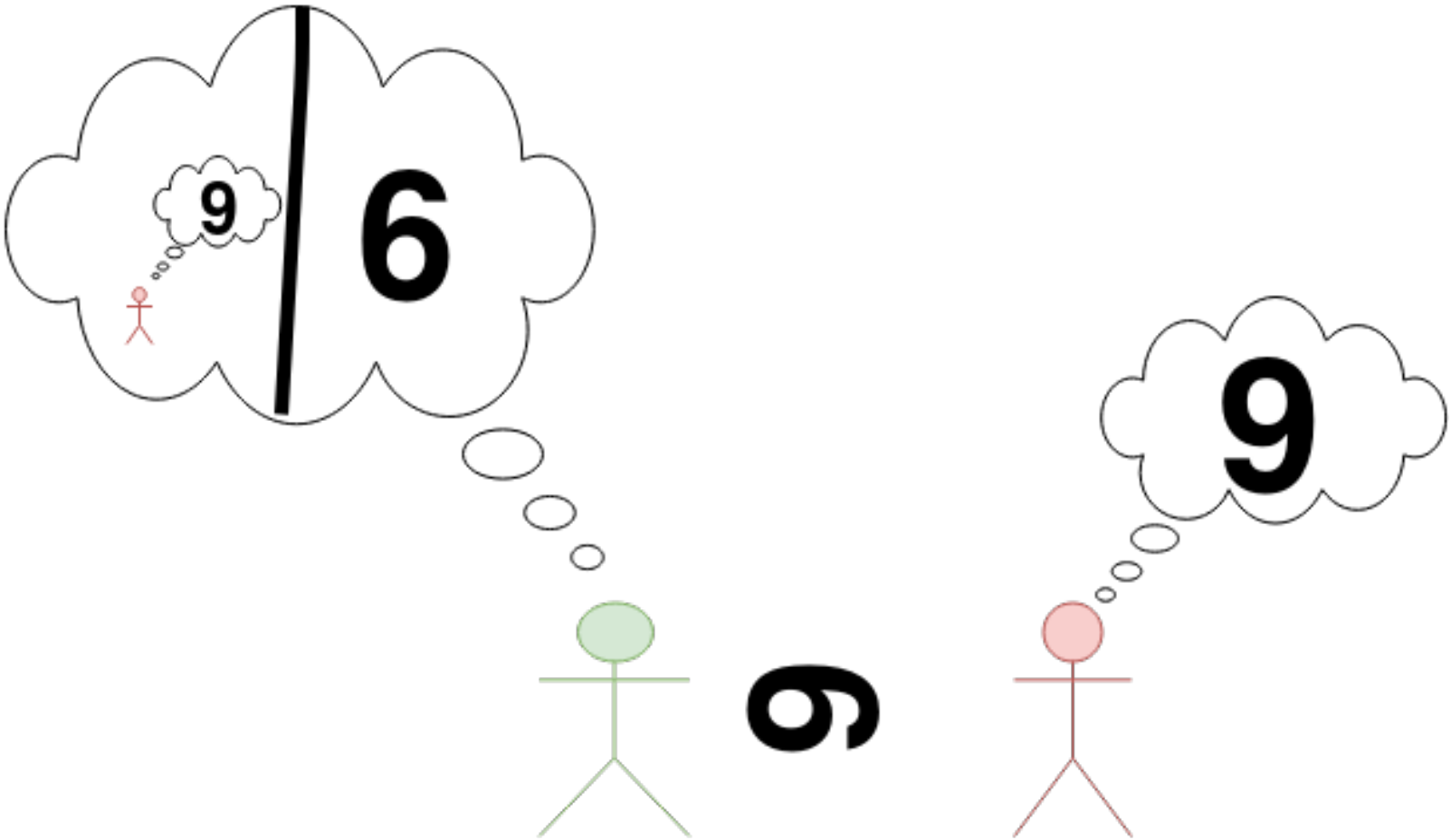


Figure 1: Theory of mind.

Experimento Sally e Anne

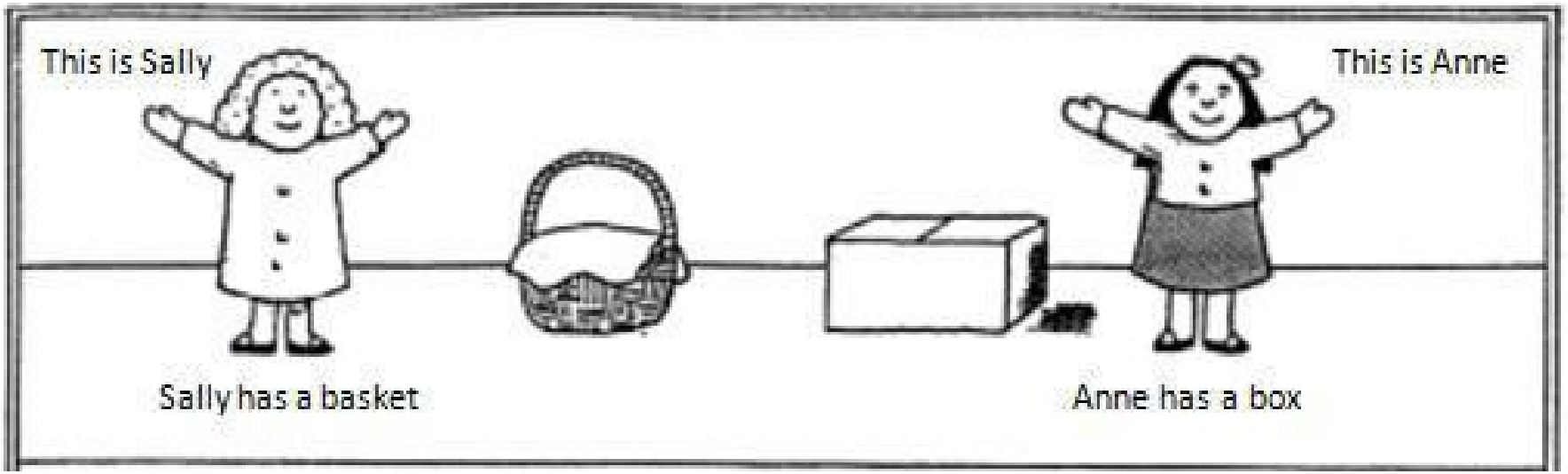


Figure 2: Experimento Sally e Anne 1.[1]

Experimento Sally e Anne

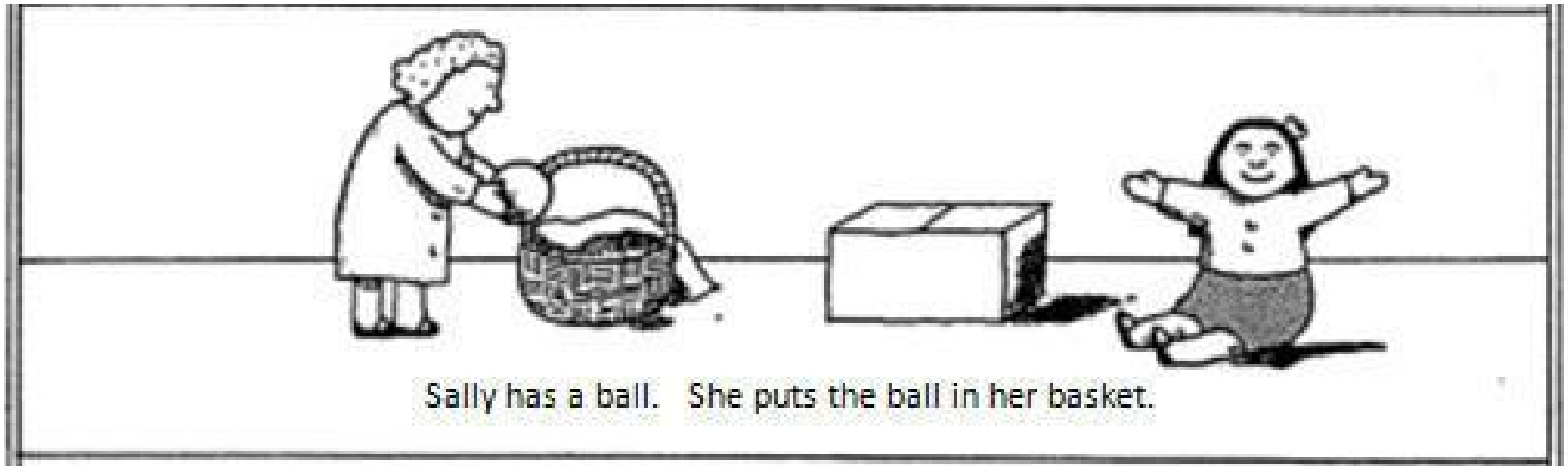


Figure 3: Experimento Sally e Anne 2.[1]

Experimento Sally e Anne

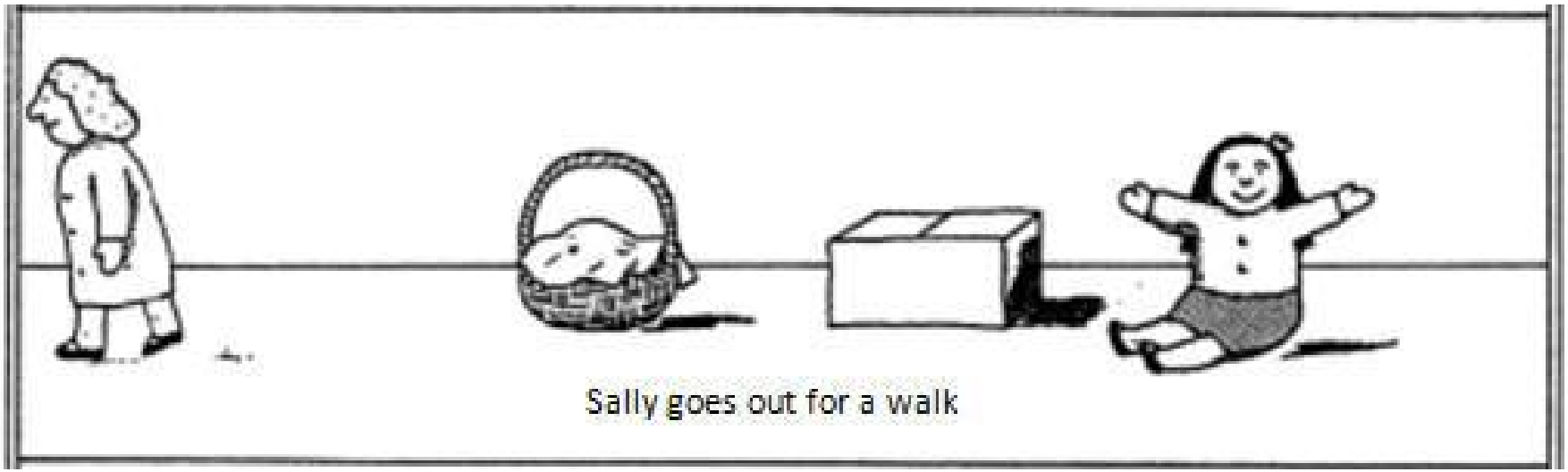


Figure 4: Experimento Sally e Anne 3.[1]

Experimento Sally e Anne

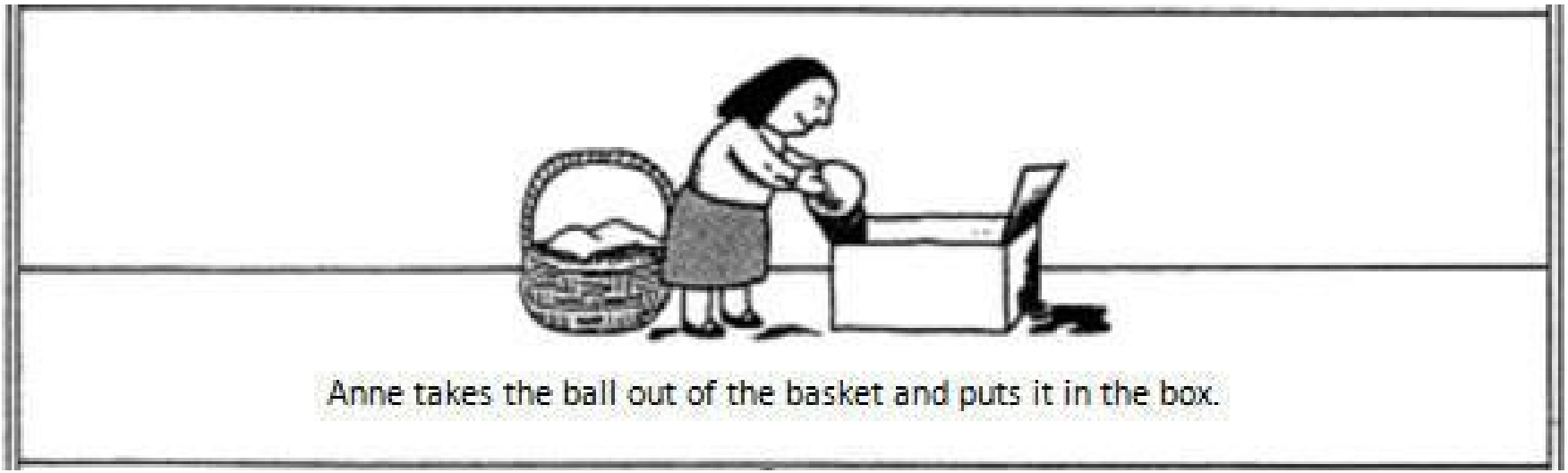


Figure 5: Experimento Sally e Anne 4.[1]

Experimento Sally e Anne

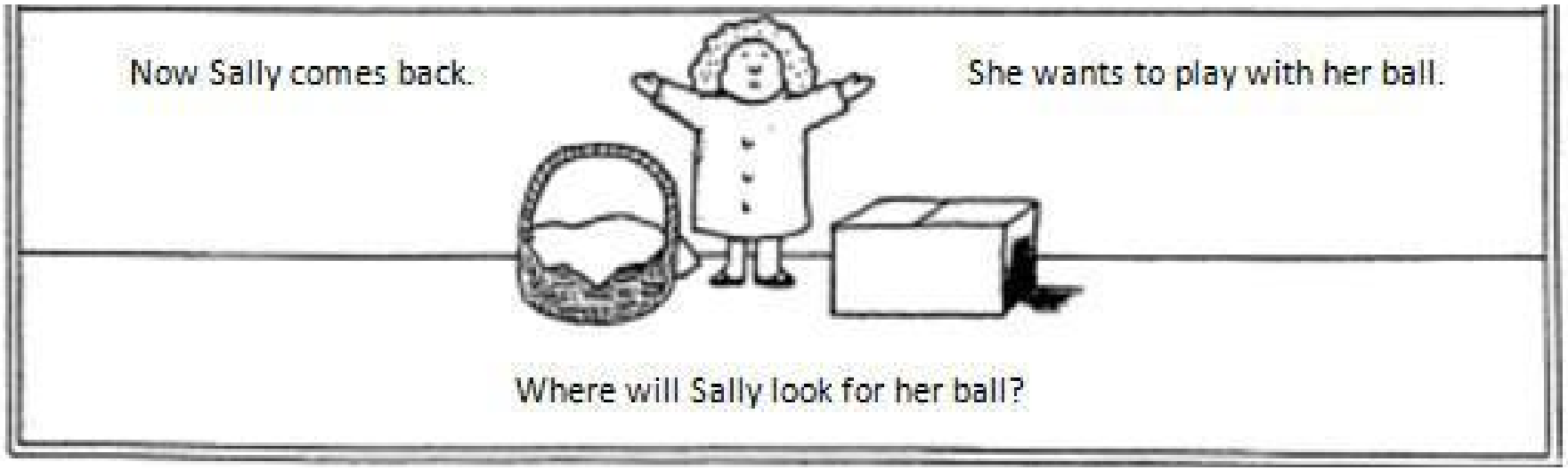


Figure 6: Experimento Sally e Anne 5.[1]

Lorini

Define uma lógica multi-modal chamada $\mathcal{L}IA$. Permitindo:

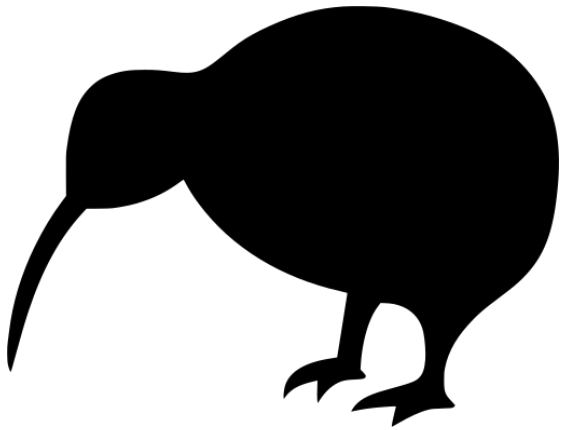
- Lidar com noção de **tentativa**, dentro de um *framework* que permite raciocinar sobre crenças, objetivos, intenções e ações básicas.
- Lidar com a geração de intenções instrumentais e entender sobre que condições um agente decide buscar um certo plano como uma forma de atingir um objetivo superior.
- Lidar com a persistência de intenções, permitindo entender sobre que condições uma intenção de executar uma certa ação pode ser abandonada.

Conceitos

- Objetivo
- Crença
- Ações
- Tentativa
- Surpresa

Crença

- Expressa ideias em que o agente acredita
- Nem sempre são fatos
  - $Belief_{Lucas}(Passaro(Titi))$
  - $Belief_{Matheus}(Passaro(Titi))$
  - $Belief_{Lucas}(Voa(Titi))$
  - $Belief_{Matheus}(\neg Voa(Titi))$



Ação

- Atuação do agente no ambiente
- Ação simples: Qualquer ação que conste no repertório do agente, e que ele consiga executar imediatamente.

Objetivo

- É um estado mental motivacional. É o que incentiva o agente a buscar algo.
- Intenção: objetivo com conteúdo acionário
  - $Goal_i(Passar_{TCC})$
- Achievement goal:
  - $AGoal_i(SeFormar)$
- Future directed Intention:
  - $FDI_i(Apresentar_{tcc}, SeFormar)$

Intenção

- Uma intenção é definida como um objetivo com conteúdo acionário. As intenções são divididas em proximais e distais.
- As intenções proximais são responsáveis por exercer controle do agente no presente.
- As intenções distais são responsáveis por guiar o comportamento e é utilizado no raciocínio do agente para geração de planos.

Tentativa

A expressão o "agente i intenciona fazer α pode ter dois significados":

- O agente i intenciona fazer α com sucesso;
- O agente i intenciona tentar fazer α.

Uma tentativa de fazer α é sempre iniciado por uma intenção proximal de tentar fazer α com sucesso ou uma intenção proximal de tentar fazer α.

- $Goal_I \ll I : Dormir \gg \top$

Surpresa

- Ocorre quando um evento não esperado pelo agente ocorre.
- Precursor de revisão de crenças

Reinforcement Learning

Uma técnica de Inteligência Artificial em que o aprendizado de agente acontece por reforço. Ou seja, pelas recompensas de suas ações.

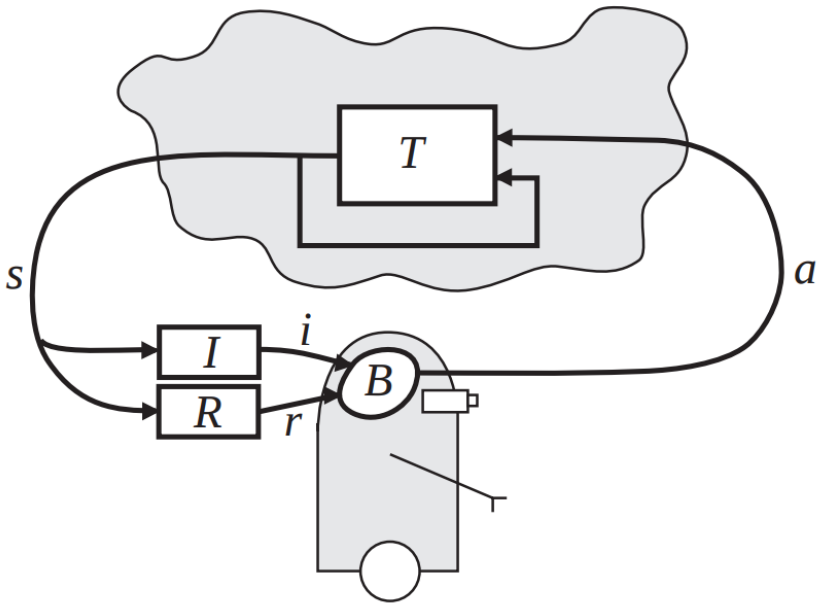


Figure 7: Modelo padrão de RL, Kaelbling et al., 1996.

DOORMAX - Deterministic Object-Oriented Rmax

DOORMAX [2] é um algoritmo baseado na estrutura do família R_{MAX}, projetado para Processo de decisão de Markov orientado à objetos (OOMDP - do inglês *Object-Oriented Markov Decision Process*) proposicional determinístico.

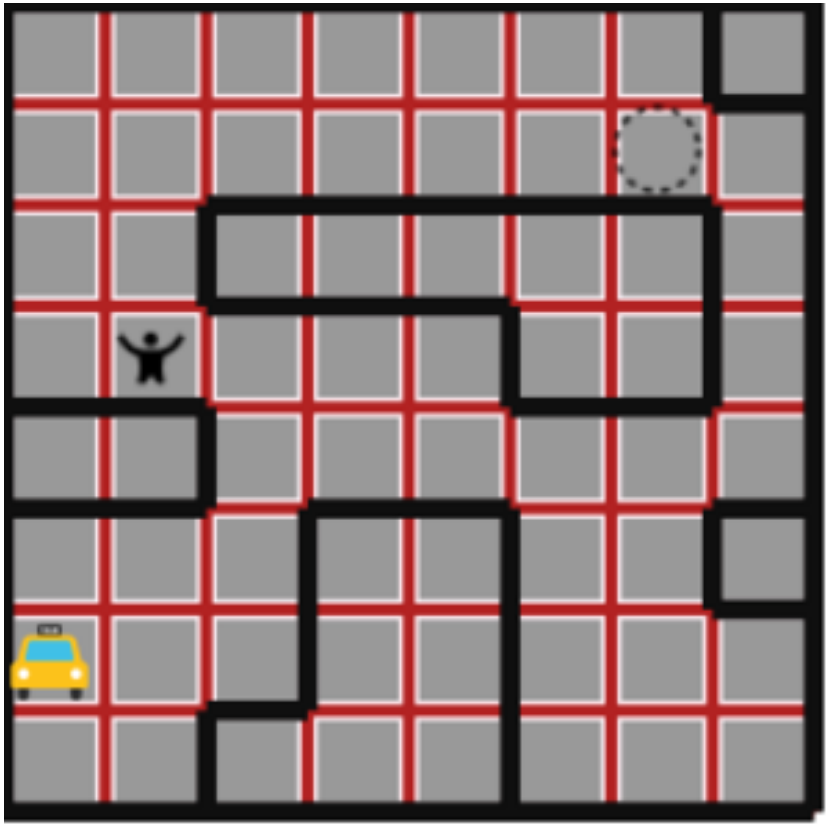
- Algoritmo gera hipóteses através do que ele percebe.
- Aprendizado de Efeitos
- Aprendizado de Condições dos Efeitos

- Efeitos $\rightarrow$ mudanças que ocorrem no mundo quando um agente executa uma ação cujas precondições estão satisfeitas.
  - Cada ação pode resultar em múltiplos efeitos.
- Precondição $\rightarrow$ São as condições necessárias para um evento acontecer quando uma determinada ação é realizada.

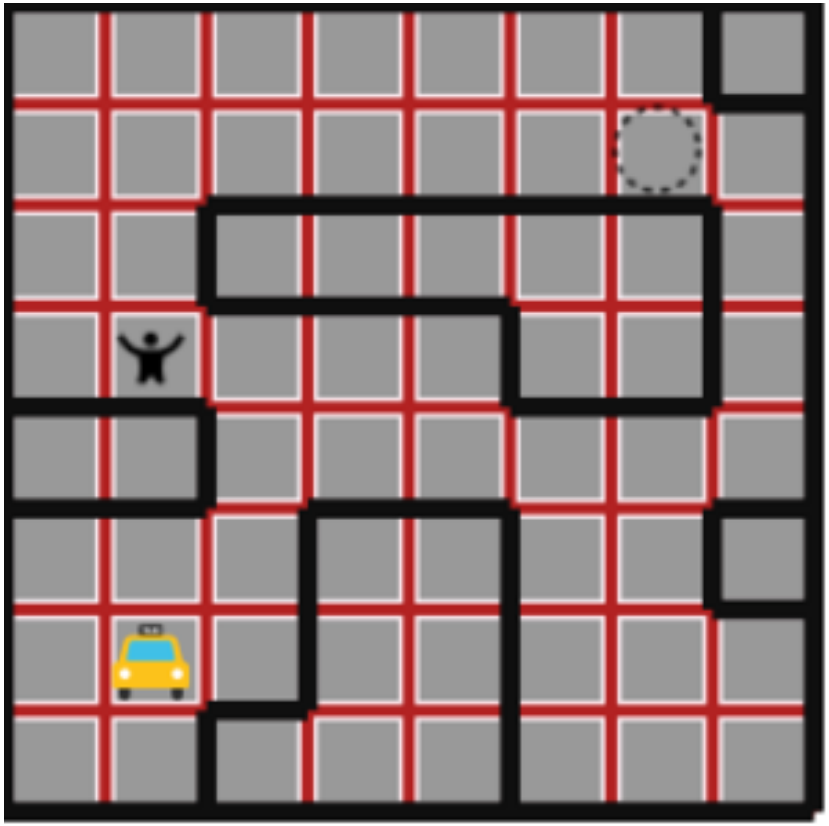
Ambiente de Testes

O ambiente escolhido para realizar os experimentos é o domínio *Taxi*. Este domínio consiste em um mundo de grid 2D, onde um agente (táxi) tem que levar um passageiro para um destino. Tanto o táxi quanto o passageiro e o destino ocupam uma posição na grid.

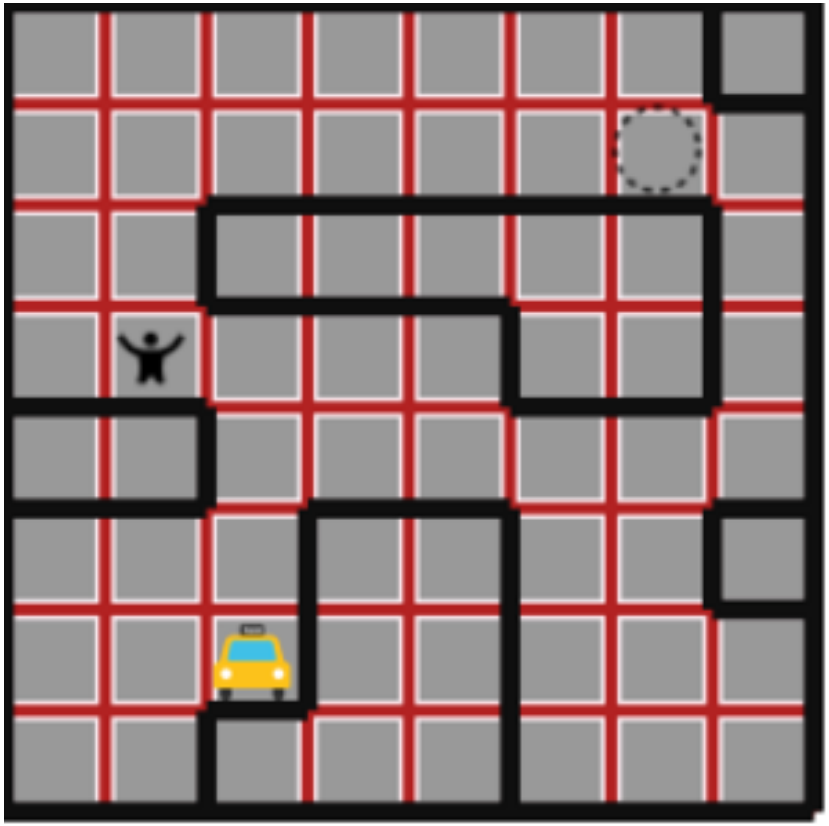
Ambiente de Testes



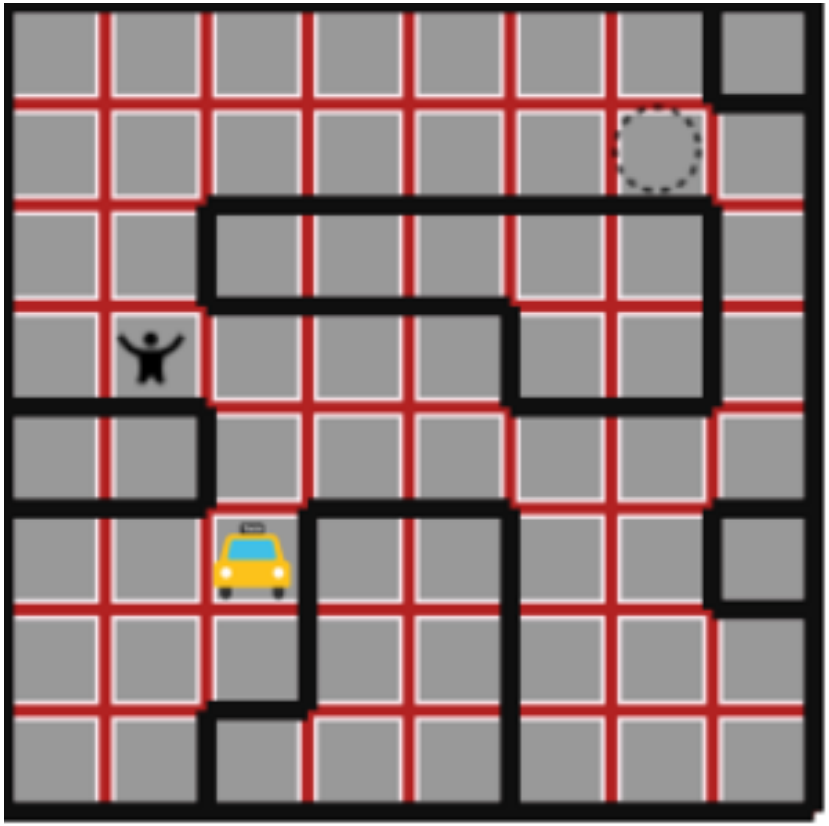
Ambiente de Testes



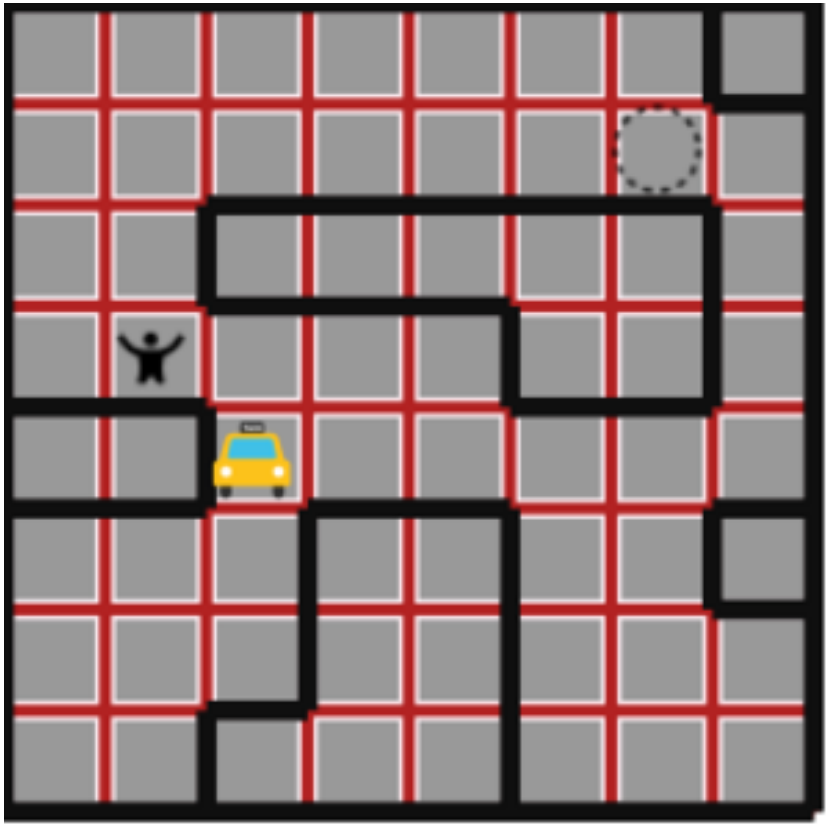
Ambiente de Testes



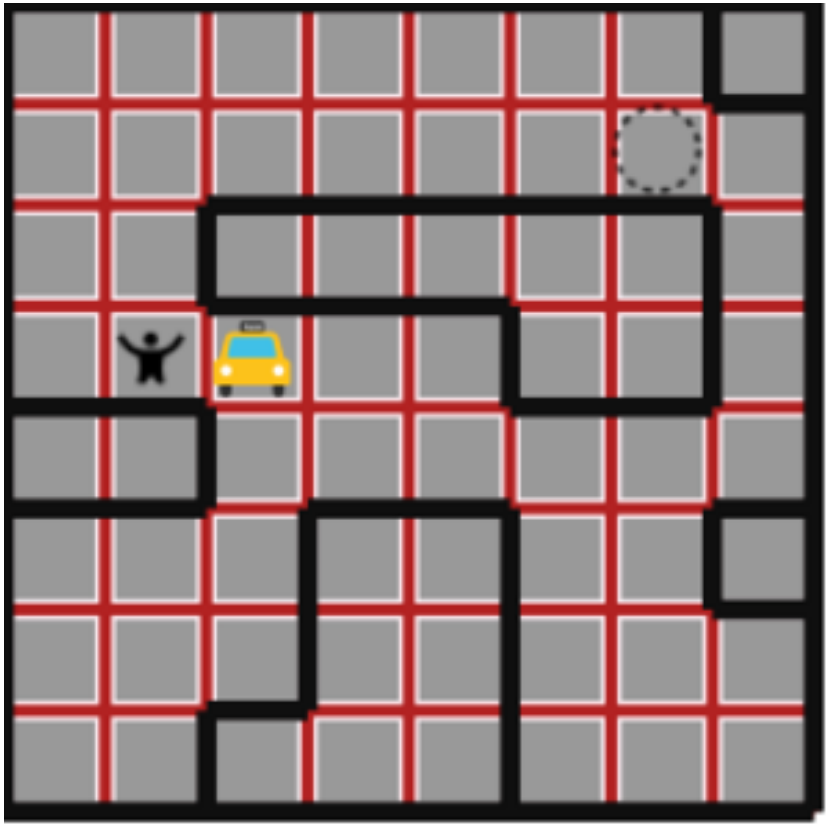
Ambiente de Testes



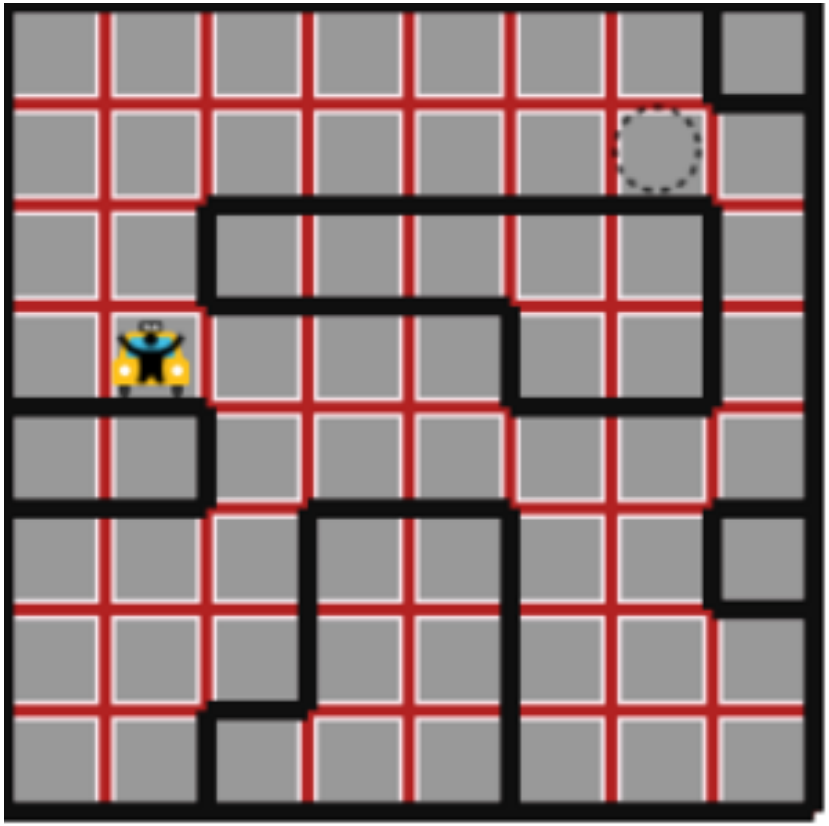
Ambiente de Testes



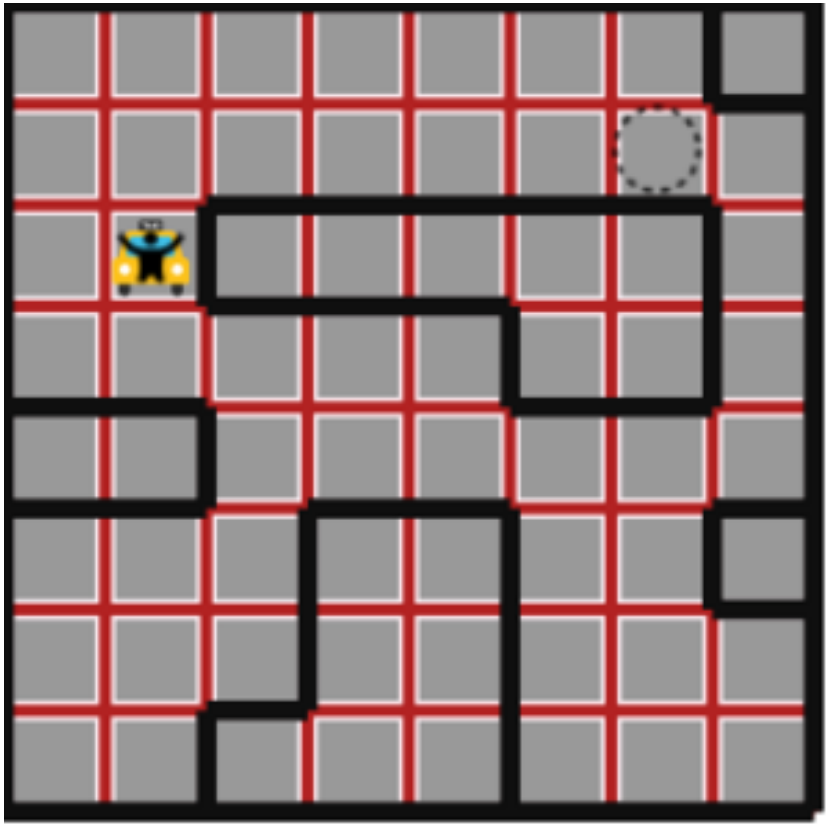
Ambiente de Testes



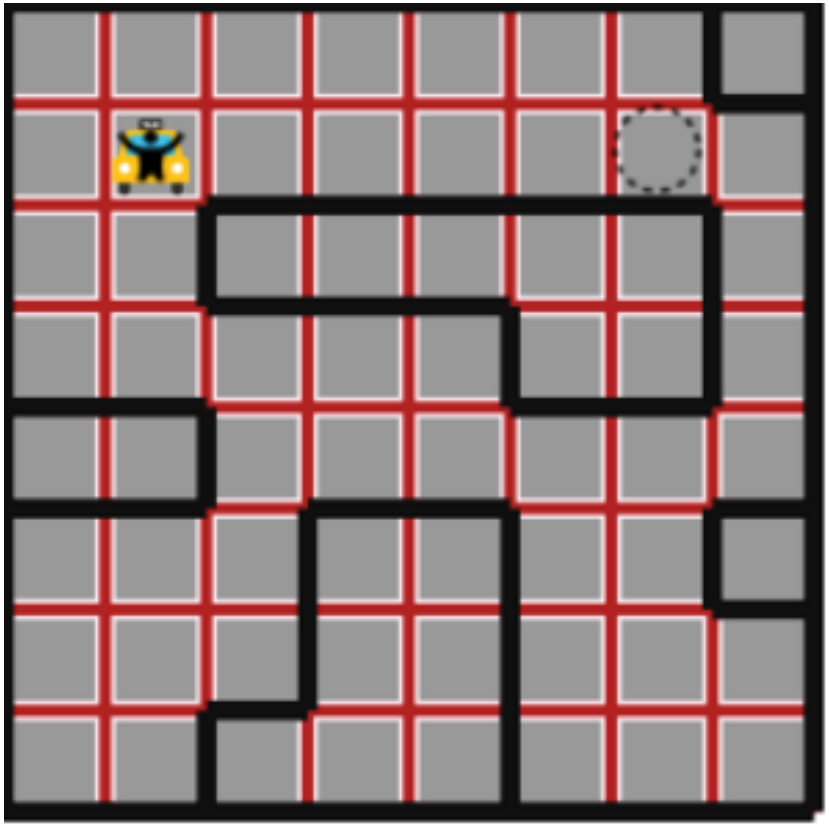
Ambiente de Testes



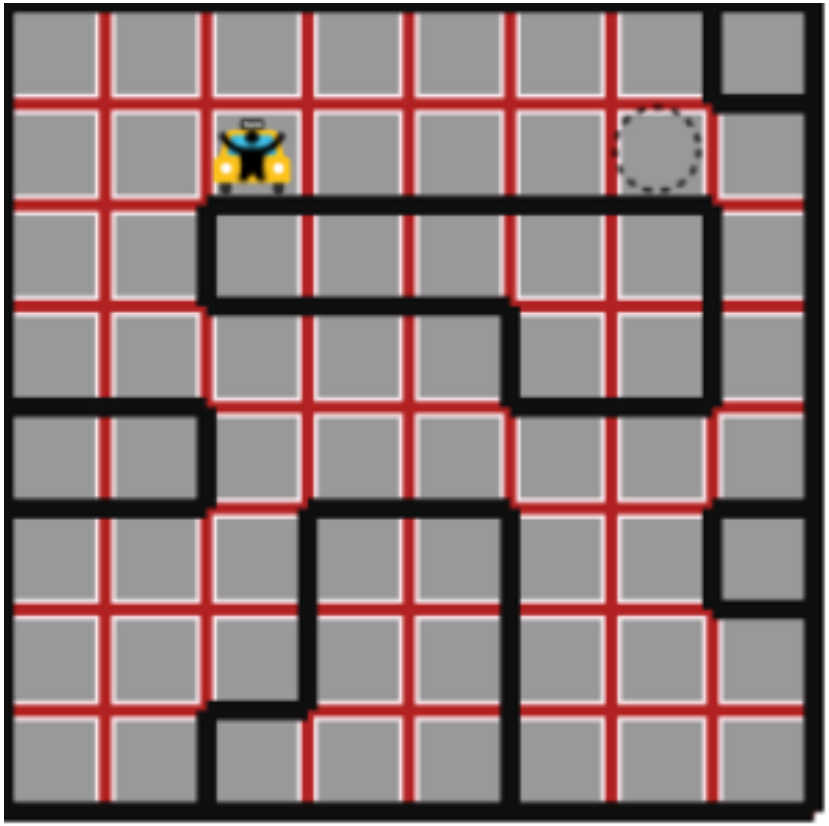
Ambiente de Testes



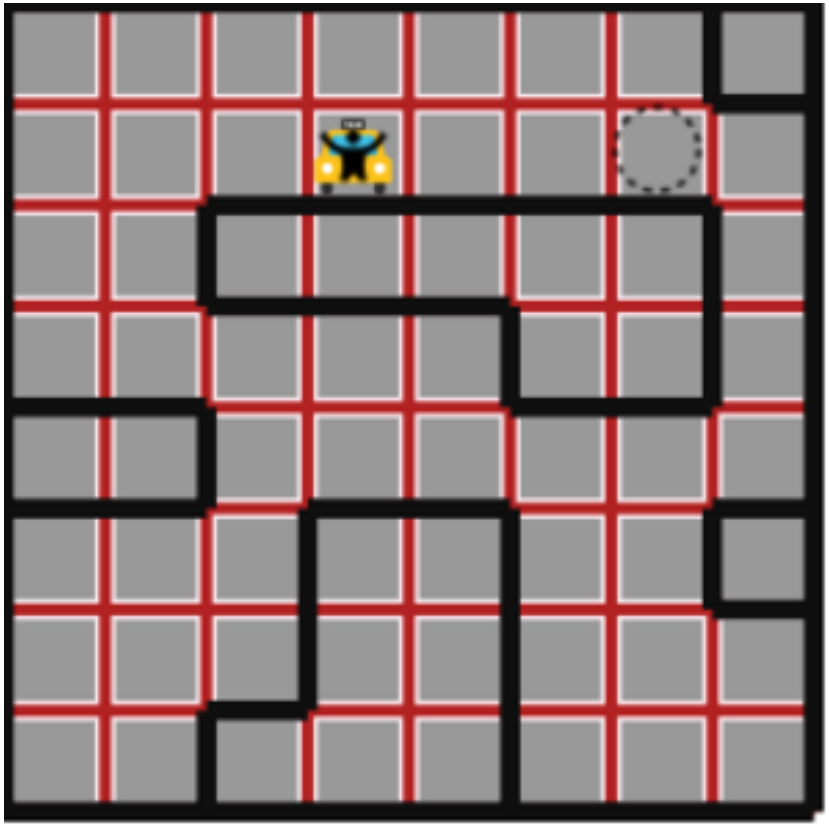
Ambiente de Testes



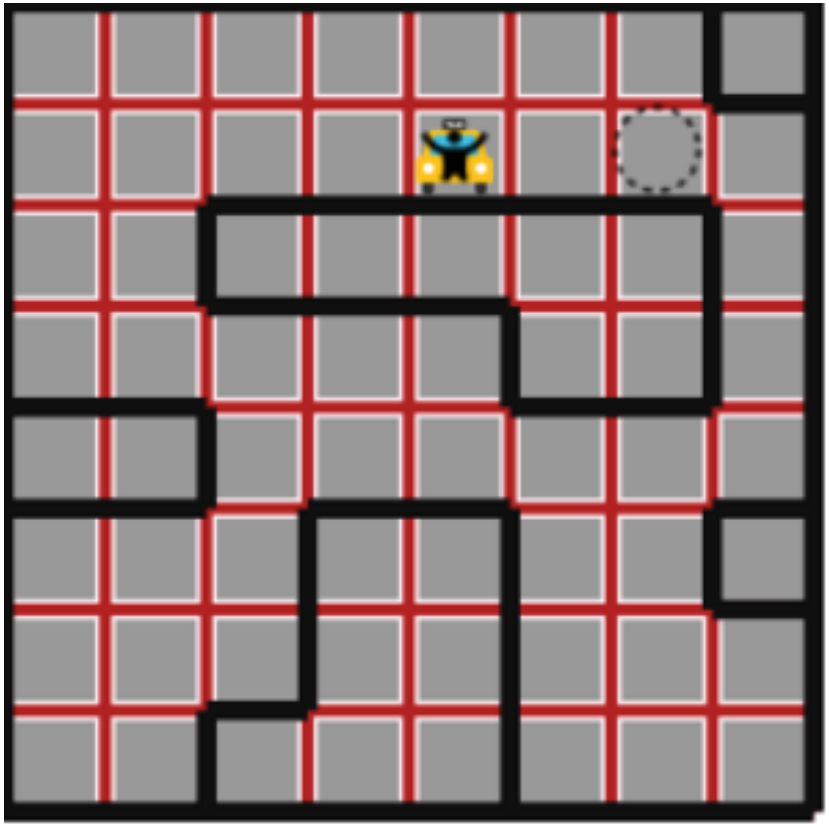
Ambiente de Testes



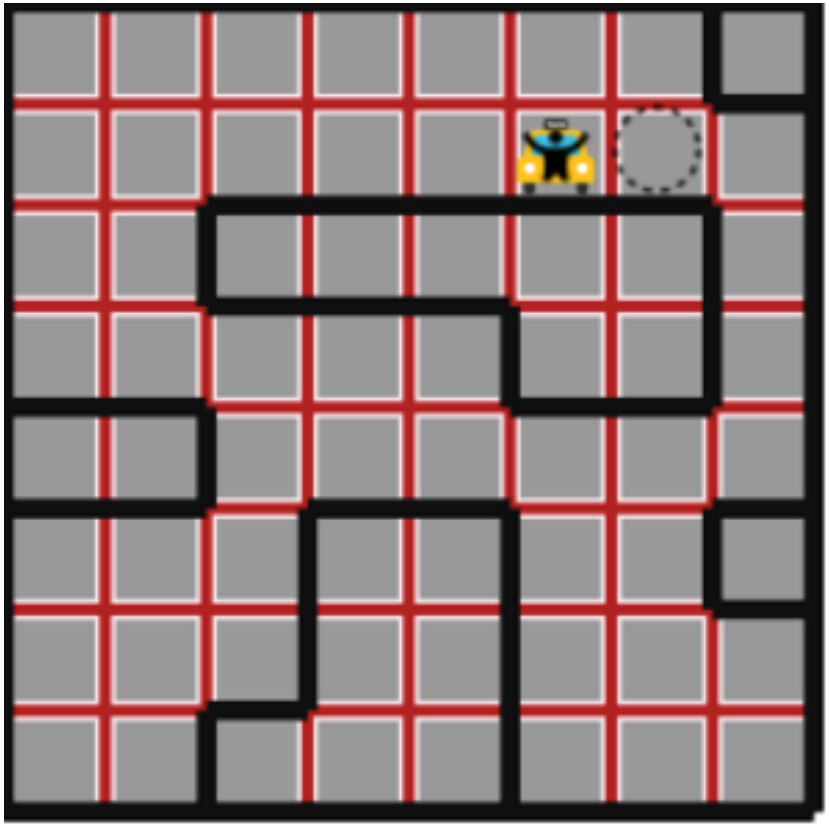
Ambiente de Testes



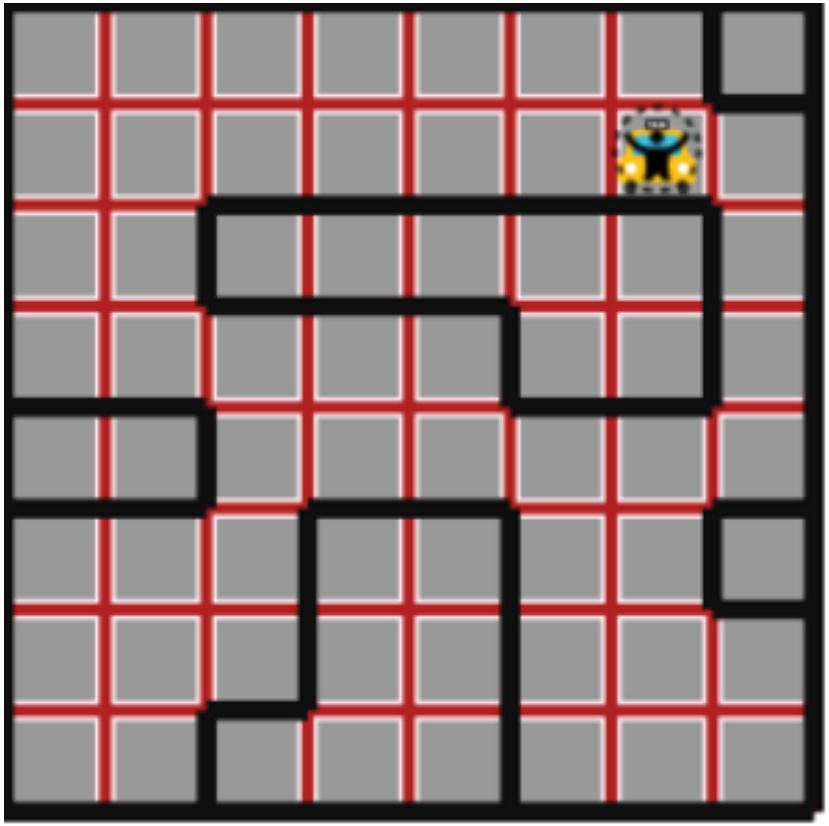
Ambiente de Testes



Ambiente de Testes



Ambiente de Testes



Ciclo de raciocínio - Geral

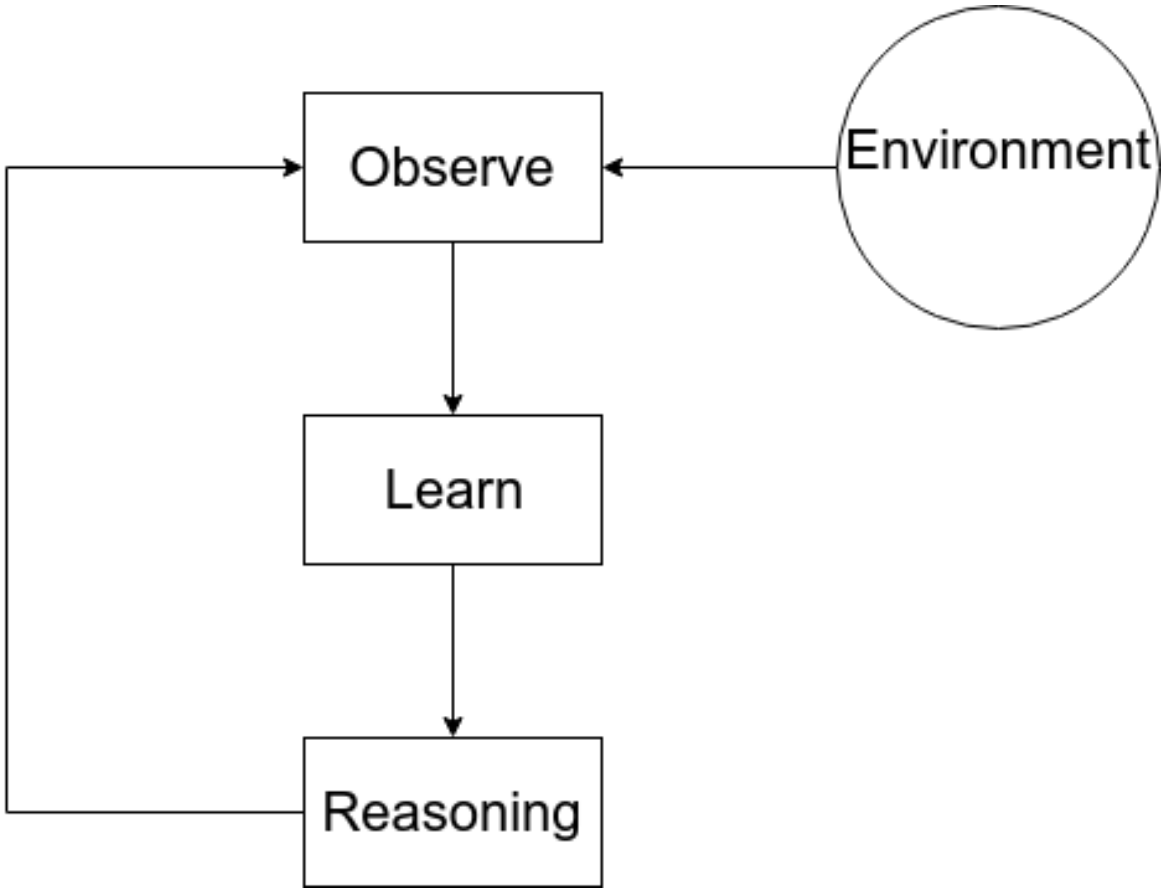


Figure 8: Arquitetura geral de um agente

Operadores Perceptuais - retornam percepções sobre o ambiente onde o agente está inserido

- *touch* $\rightarrow$ informações referentes a casas adjacentes;
- *on* $\rightarrow$ informações referentes a casa atual do agente;
- *see* $\rightarrow$ informações referentes a demais casas do tabuleiro;

Ciclo de raciocínio - Agente agindo no mundo

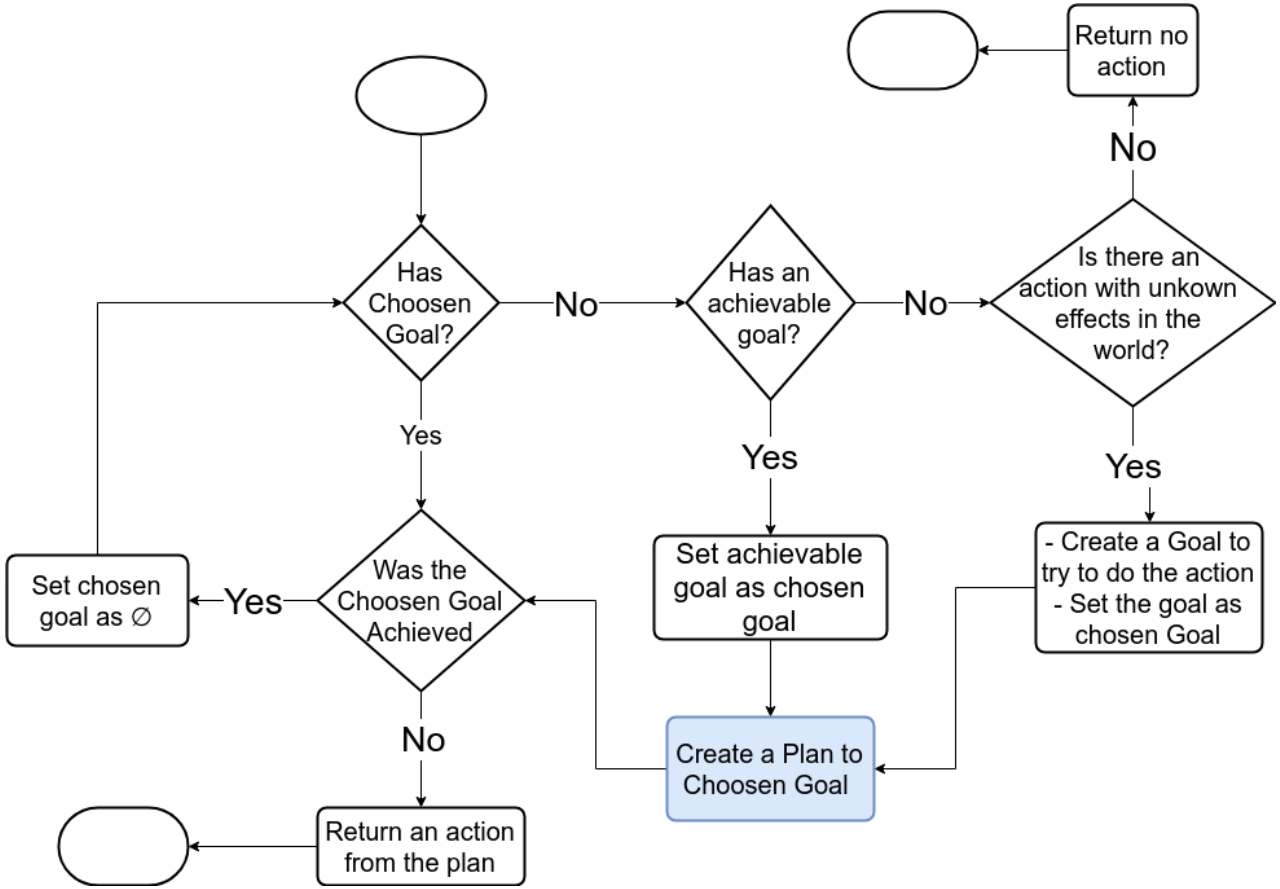


Figure 9: Ciclo de raciocínio de um agente agindo no mundo.

Ciclo de raciocínio - ToM

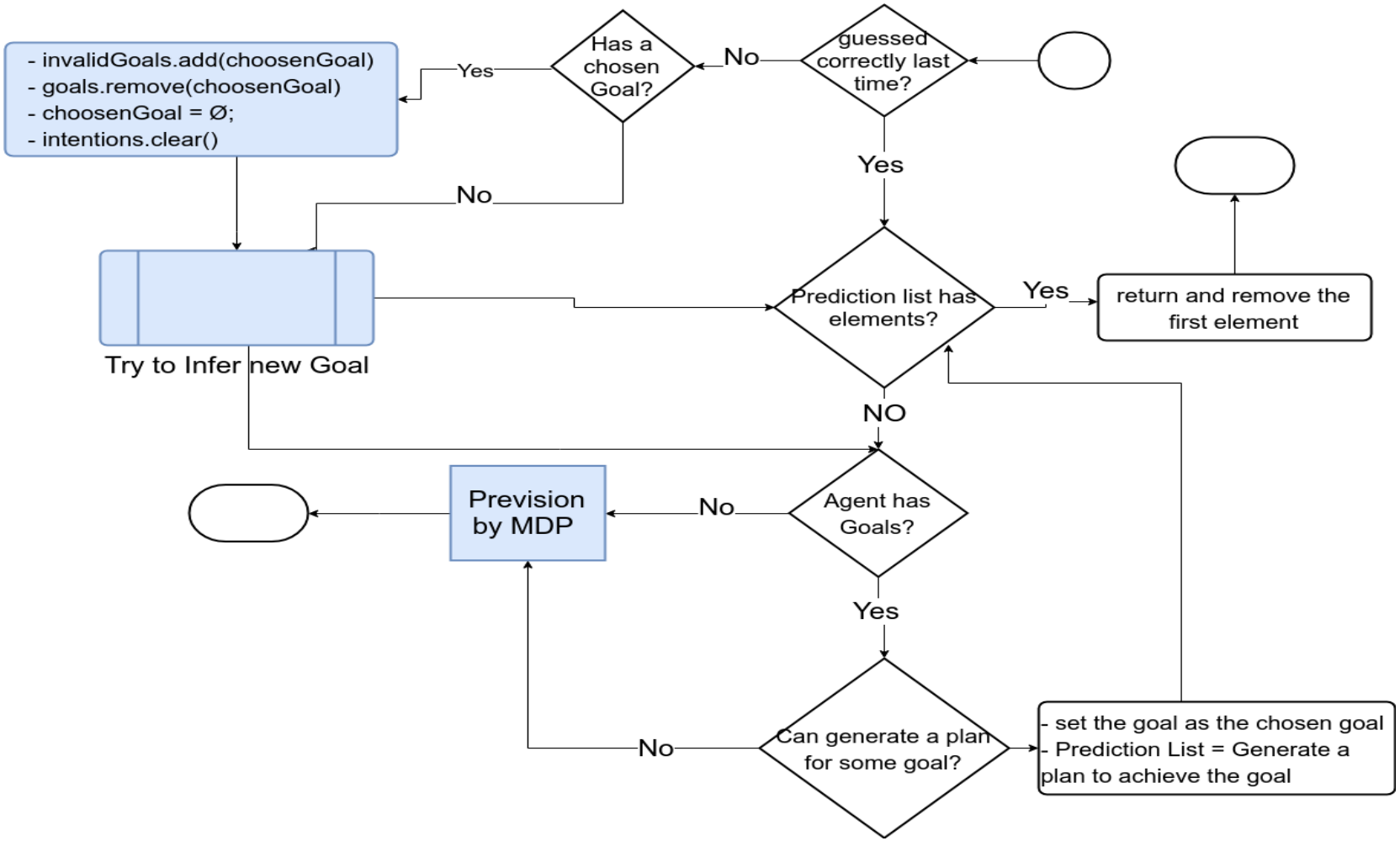


Figure 10: Ciclo de raciocínio ToM

Experimentos

1. O agente é capaz de agir e aprender sobre o mundo ao seu redor?
2. O agente é capaz de aprender sobre outros agentes, e predizer suas ações?
3. O agente é capaz de revisar suas próprias crenças, quando observando outros agentes?

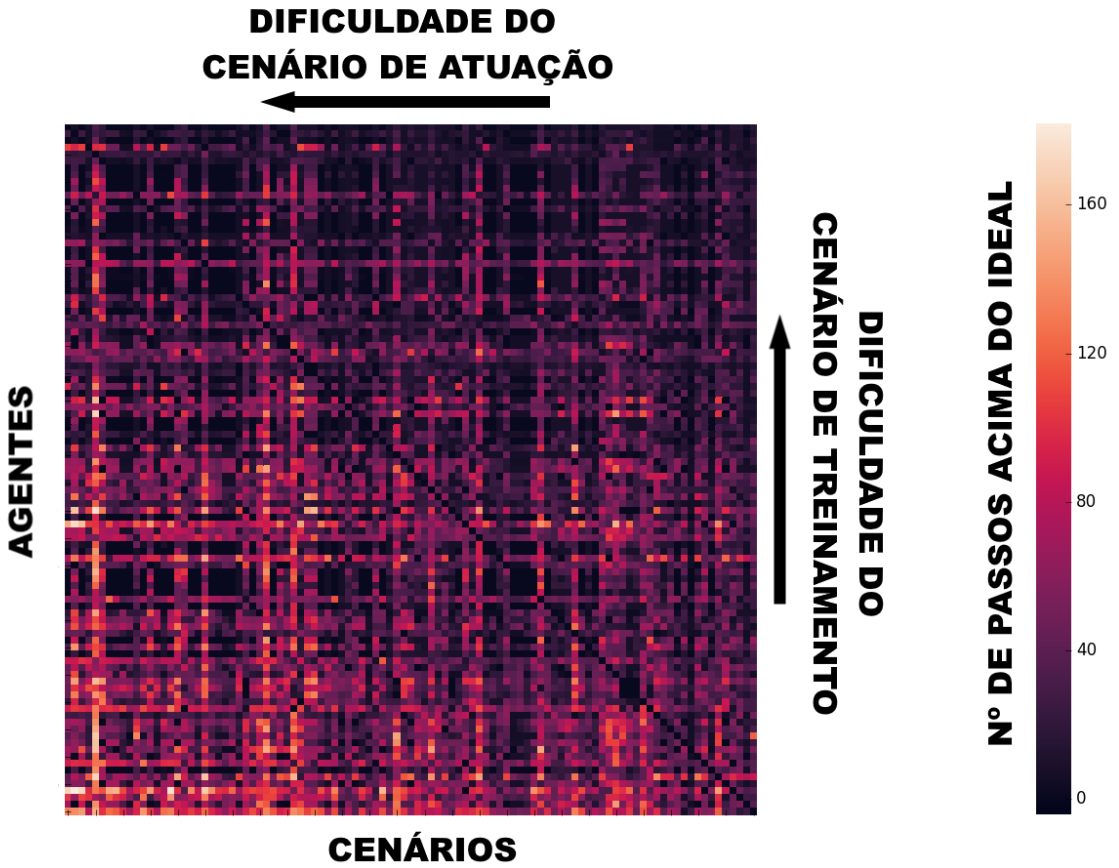
Experimento 1

O agente é capaz de agir e aprender sobre o mundo ao seu redor?

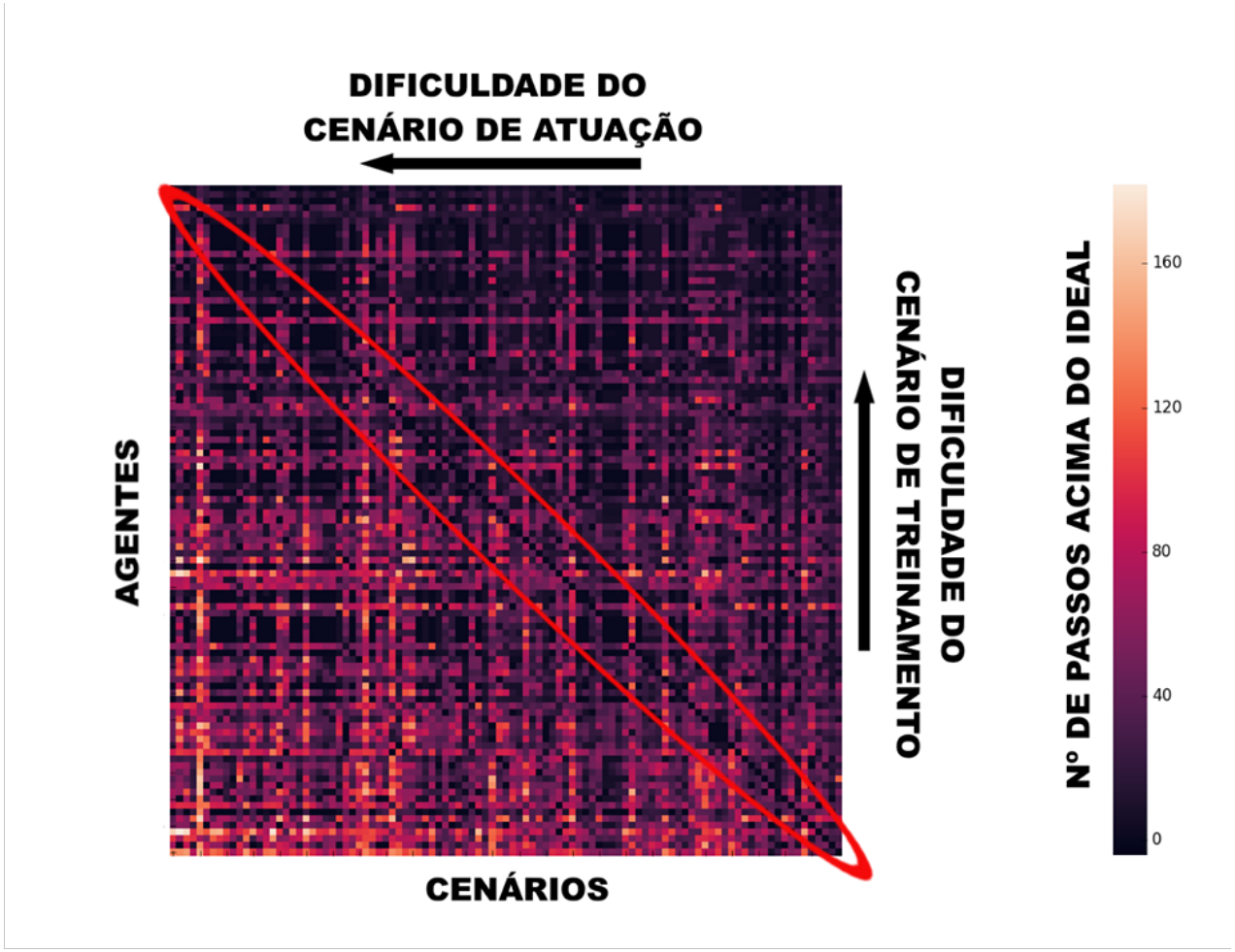
- Foram gerados 100 cenário de teste;
- 100 agentes diferentes;
- Todos agiram nos 100 cenários;
- Comparação com número de passos ideal.

Os agentes que aprenderam em cenários mais difíceis tem um desempenho melhor?

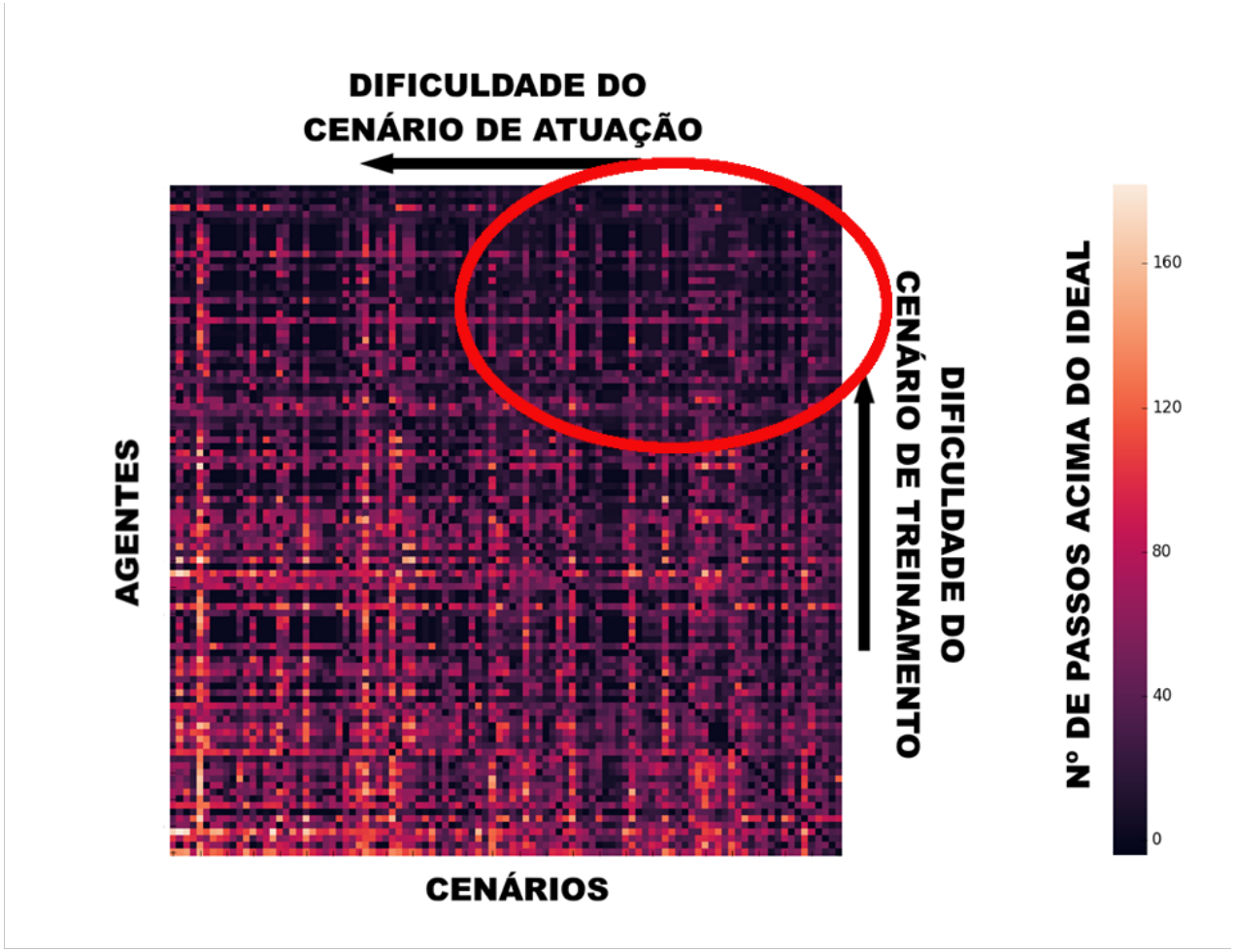
Experimento 1



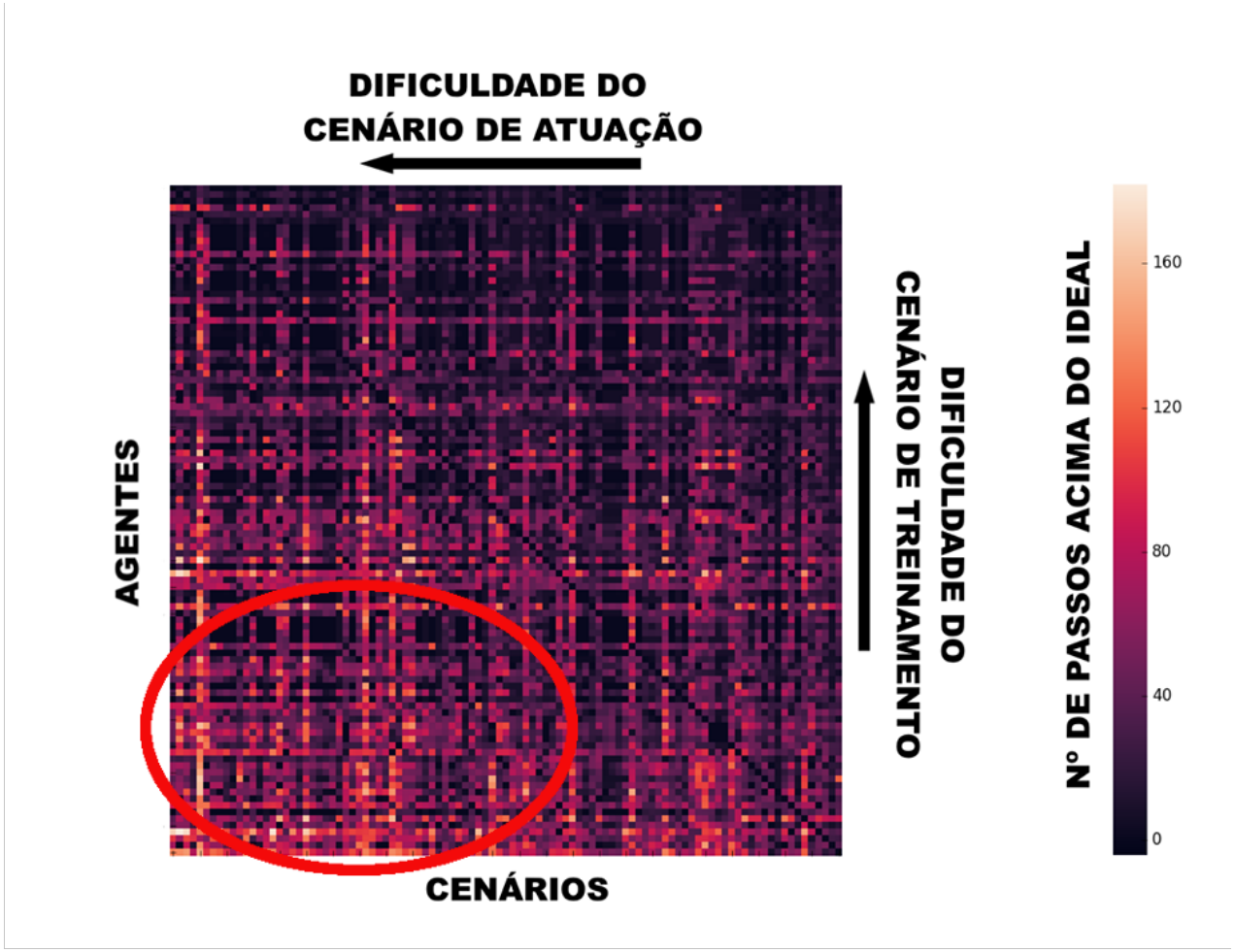
Experimento 1



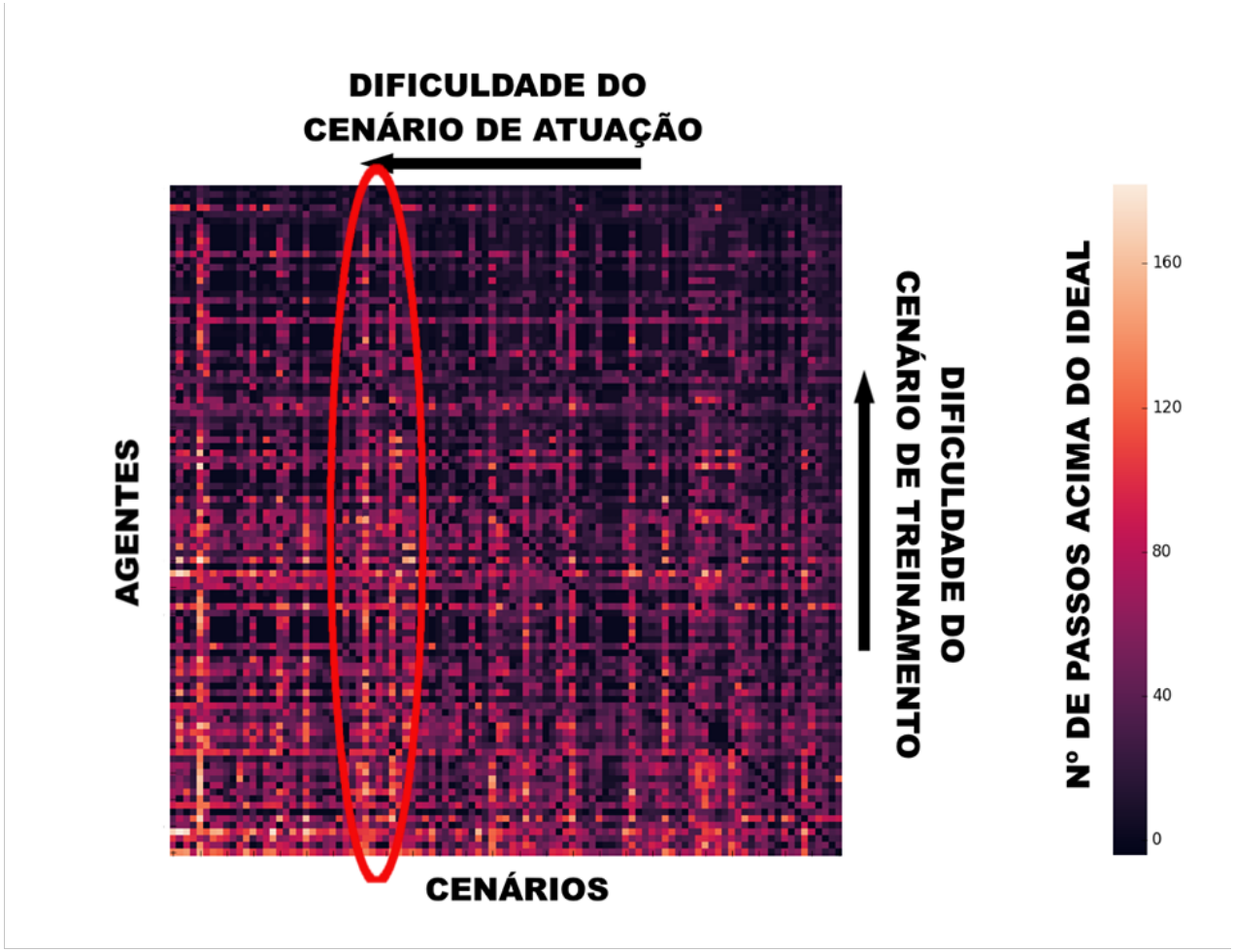
Experimento 1



Experimento 1



Experimento 1

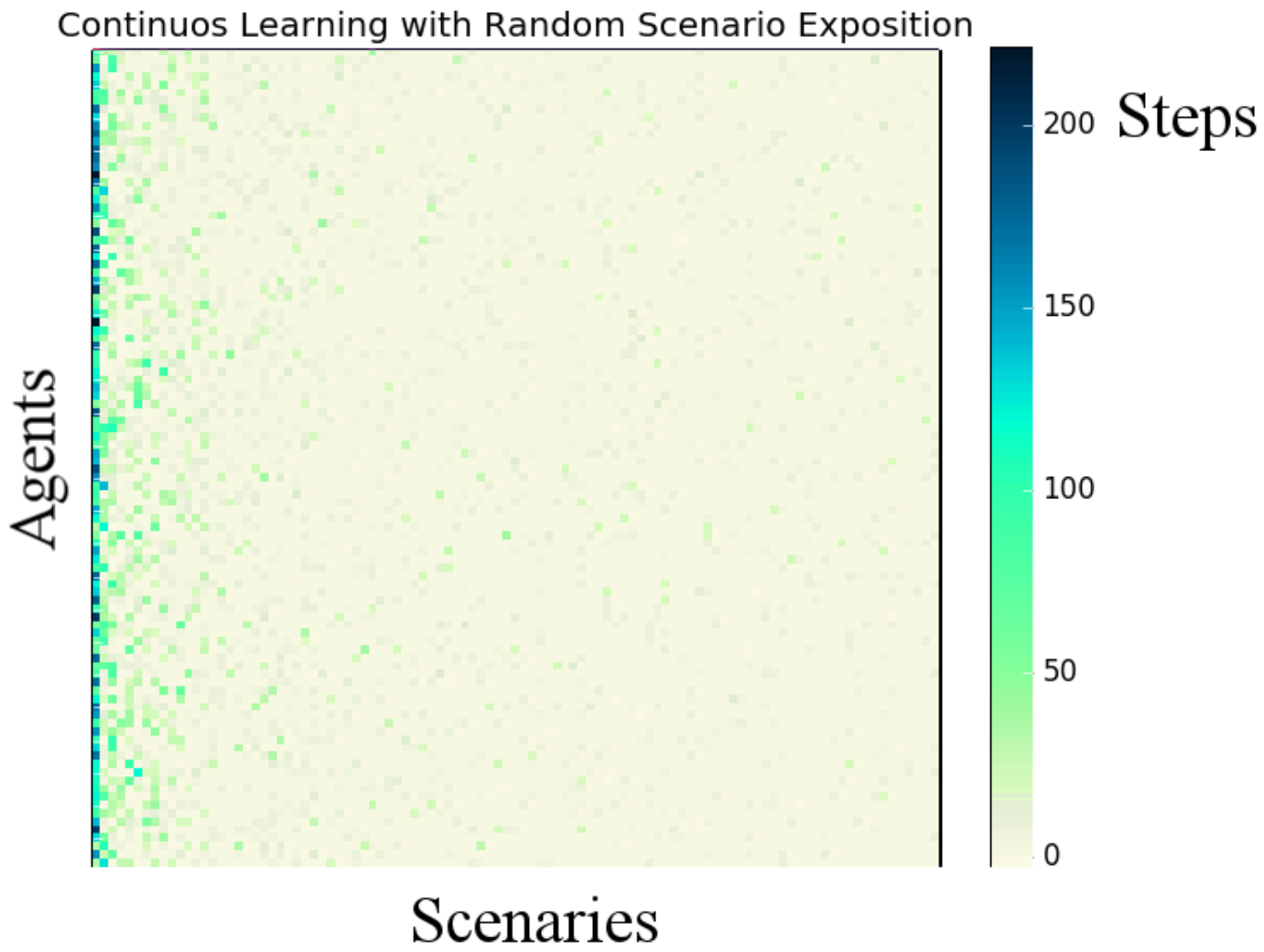


Experimento 1

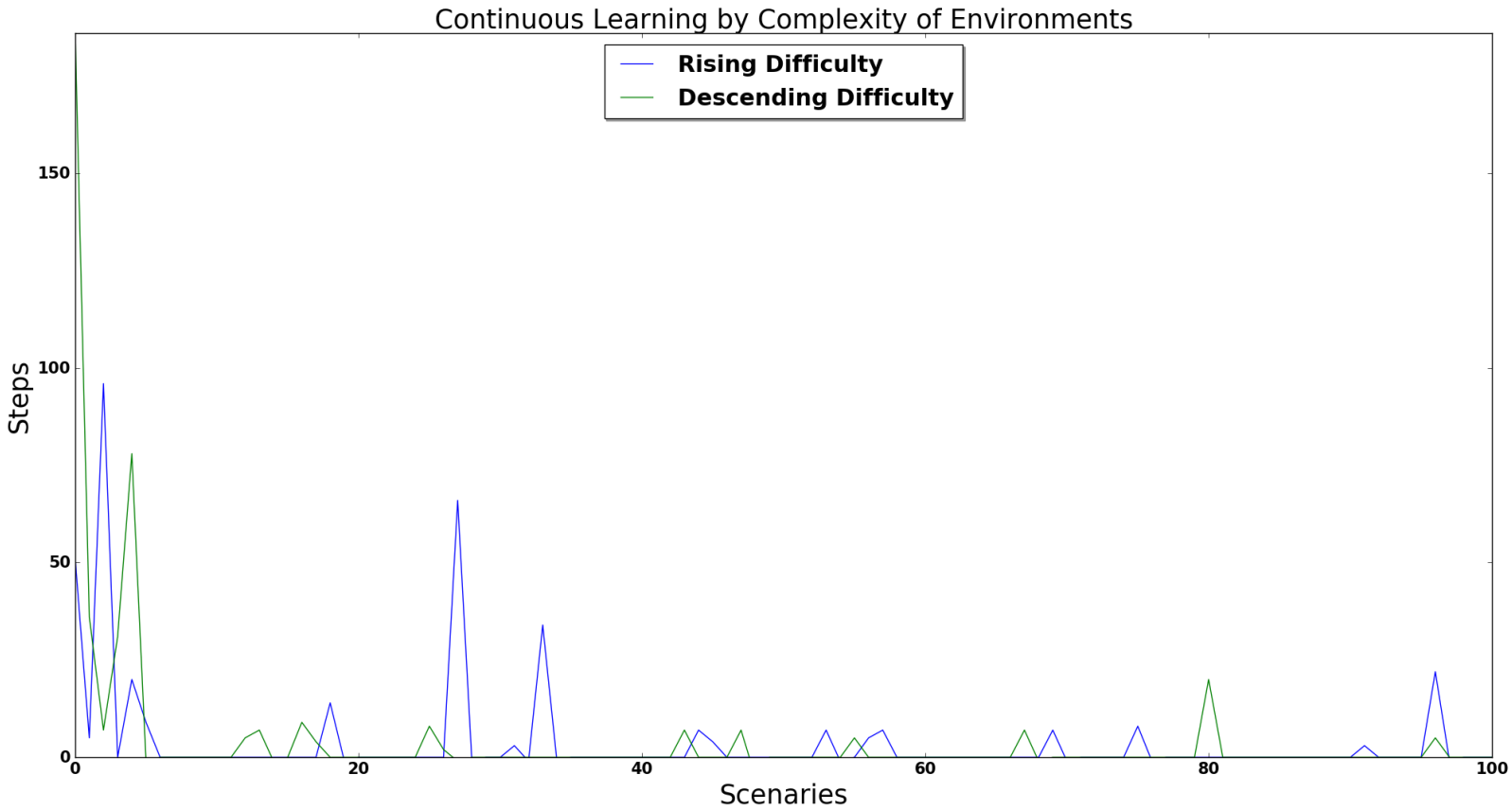
- Agora com aprendizado persistente;
- 100 agentes;
- 100 cenários em ordem aleatória.

-Os agentes terão um desempenho progressivamente melhor?

Experimento 1



Experimento 1



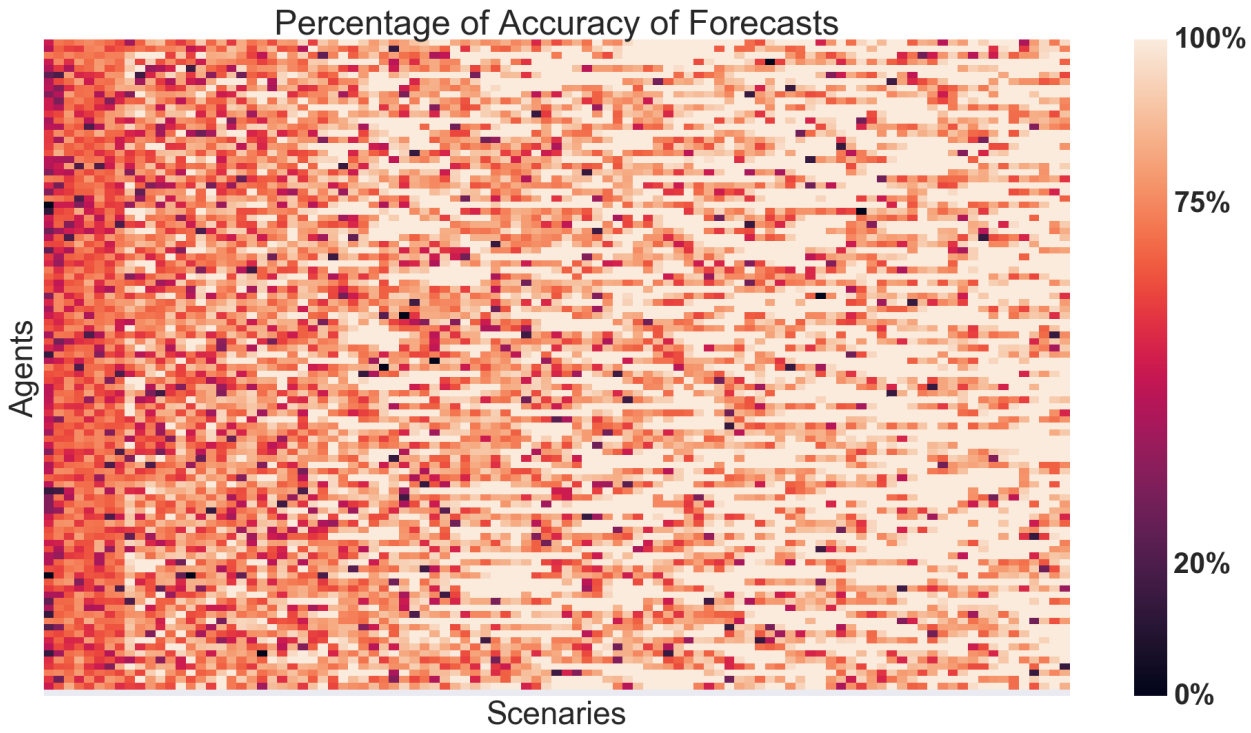
Experimento 2

O agente é capaz de aprender sobre outros agentes, e predizer suas ações?

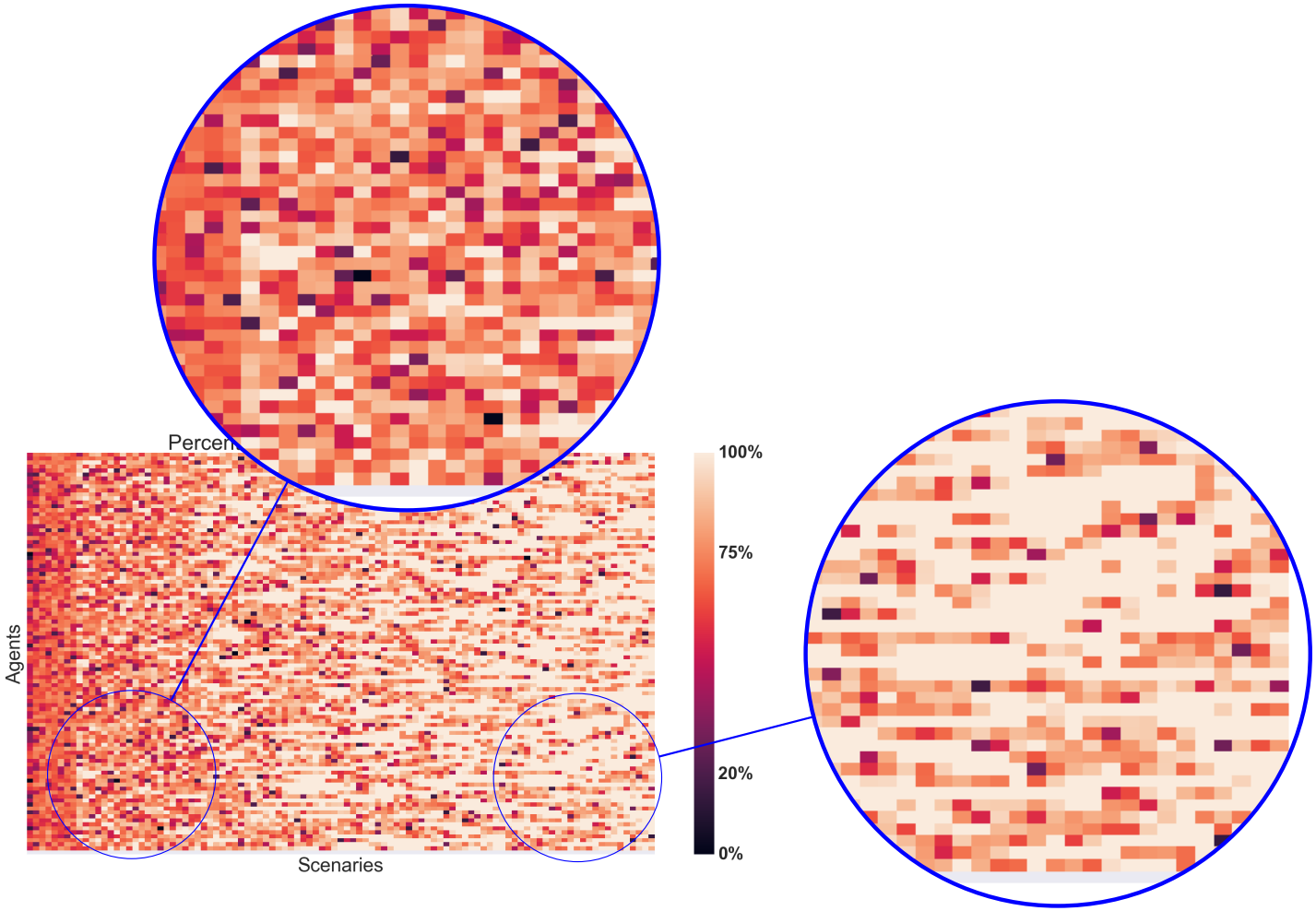
- Os mesmos 100 cenários foram usados.
- Foram gerados 100 agentes sem conhecimentos prévios
- Apenas observam ações de outro agente treinado
- Ordem aleatória

  Os agentes fazem progressivamente melhores previsões?

Experimento 2



Experimento 2



Experimento 3

Os 100 cenários foram divididos em um subconjuntos de 50 cenários de treino e 50 de teste. O desempenho de três agentes são treinados por experimentação em um dos cenários de treino.

- O primeiro será treinado somente no primeiro cenário;
- O segundo treina também nos outros 50 cenários por experimentação;
- O terceiro observa outro agente nos outros 49 cenários.

Experimento 3

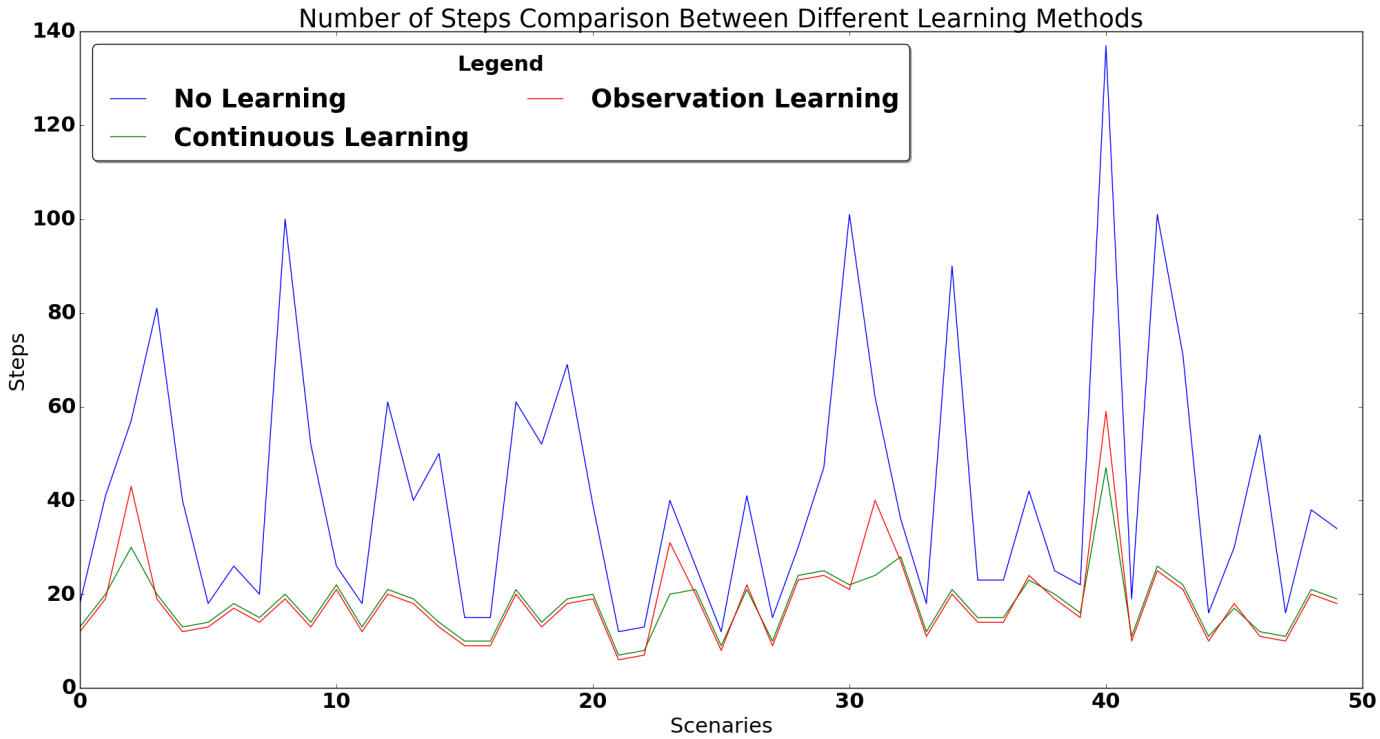


Figure 12: Desempenho dos três agentes.

Exemplos Saída lógica - Operadores Perceptuais

- $Belief_l(\neg Touch(taxi, North))$
- $Belief_l(See(horizontalWall(leftStartOfWall : 0, rightStartOfWall : 4, wallOffSet : 0)))$

Exemplos Saída lógica -

- $Belief_I(\gamma^+(taxiMoveNorth, (passenger, yLocation, ArithmeticEffect, -1)) \rightarrow \neg Touch(taxi, North) \wedge \neg Touch(taxi, South) \wedge On(taxi, passenger) \wedge In(taxi, passenger))$
- $Belief_I(\gamma^+(taxiMoveNorth, (taxi, yLocation, ArithmeticEffect, -1)) \rightarrow \neg Touch(taxi, North))$

Saída lógica

- $AGoal_l{<}I$: $dropOffPassenger{>}(\gamma^+(taxi, passengerInTaxi, AssignmentEffect, 0) \wedge \gamma^+(passenger, inTaxi, AssignmentEffect, 1)$
- $AGoal_l{<}I$: $pickupPassenger{>}(\gamma^+(taxi, passengerInTaxi, AssignmentEffect, 1) \wedge \gamma^+(passenger, inTaxi, AssignmentEffect, 1))$
- $Goal_l \ll I : dropOffPassenger \gg \top$
- $FDI_l{<}I$: $dropOffPassenger{>}(\gamma^+(taxi, passengerInTaxi, AssignmentEffect, 0) \wedge \gamma^+(passenger, inTaxi, AssignmentEffect, 0))$

Conclusões

Esse trabalho mostrou uma modificação lógica e operacional da lógica de Lorini (2007), atráves da incorporação de aprendizado por reinforment learning, com as suas crenças explicitadas.
Foi demonstrado uma abordagem com processo de aprendizado genérico, em que muitas das etapas do ciclo de raciocínio podem ser facilmente trocados para atender a uma gama variada de cenários.

Conclusões

Nosso agente atendeu as expectativas nos três experimentos realizados, sendo capaz de aprender por experimentação, atribuir crenças a outros agentes por observação e revisar suas próprias crenças quando necessário.

Conclusões e Atividades Futuras

Etapas de geração de intenção e planejamento implementadas foram implementadas de uma forma simples, que apenas atendesse ao escopo dos experimentos. Essas etapas podem ser revisadas em trabalhos futuros para melhorar o desempenho do agente.

Obrigado pela atenção!

Referências

"The original Sally-Anne illustration by Uta Frith," 1985.

C. Diuk, A. Cohen, and M. L. Littman, "An object-oriented representation for efficient reinforcement learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 240–247, ACM, 2008.

KWIK

Know what it knows (KWIK) é um tipo de algoritmo que realiza apenas predições exatas, retornando "eu não sei" ($\perp$) quando não consegue predizer o resultado de um evento. O número de vezes em que o algoritmo retorna $\perp$ deve ser limitado.

Rmax

O algoritmo R_{max} é um KWIK que opera atribuindo a máxima recompensa possível para cada estado desconhecido. Ou seja, o algoritmo tem um viés otimista sob incerteza.

Modelo - Definições do Ambiente

O modelo de objetos é definido nos conceitos de classes, atributos, domínio, objetos, instâncias, estados, relações entre classes e efeitos.

Predicados não necessariamente precisam ser associados com uma posição no espaço, e podem representar outros tipos de conhecimento sobre objetos no mundo, como a quantia de energia de um determinado agente. Eles são sempre relações binárias.

MDP

- S = Conjunto de estados;
- A = Conjunto de ações;
- $T : S \times A \rightarrow PD(S)$ = Funções de transição;
- $R : S \times A \rightarrow \mathcal{R}$ = Função de recompensa;
- γ = Fator de desconto.

Markov Table Example

	North	East	South	West	Pick	Drop
North	6	1	0	1	0	0
East	0	4	0	0	1	0
South	0	0	0	0	0	0
West	0	0	0	2	0	1
Pick	1	0	0	0	0	0
Drop	0	0	0	0	0	0

FDI

Inferência de Intenção

Mais saídas lógicas

- $Belief_l(See(Myself(\neg passengerInTaxi, xLocation : 1, yLocation : 3)))$
- $Belief_l(See(passenger(\neg inTaxi, xLocation : 3, yLocation : 0)))$

Saída lógica

- $Belief_I(\gamma^+(taxiMoveSouth, (taxi, yLocation, ArithmeticEffect, 1)) \rightarrow \neg Touch(taxi, South))$
- $Belief_I(\gamma^+(pickupPassenger, (taxi, passengerInTaxi, AssignmentEffect, 1)) \rightarrow Touch(taxi, North) \wedge \neg Touch(taxi, South) \wedge \neg Touch(taxi, West) \wedge Touch(taxi, East) \wedge On(taxi, passenger) \wedge \neg In(taxi, passenger) \wedge \neg On(taxi, destination))$