AgentSpeak(Py)
Interpretador de AgentSpeak(L) em Python

André Luiz Leonhardt dos Santos

Orientador: Prof. Dr. Felipe Meneguzzi

Curso de Sistemas de Informação - Pontifícia Universidade Católica do Rio Grande do Sul

✉ andre.santos.004@acad.pucrs.br

Motivação

- Sistemas multiagentes são utilizados para a construção de sistemas massivamente distribuídos;
- Linguagens surgiram para facilitar o desenvolvimento destes sistemas, entre elas o AgentSpeak(L);
- O uso do AgentSpeak(L) em cenários com uma complexidade elevada requer um interpretador enxuto e eficiente;
- Neste trabalho, desenvolvemos um interpretador da linguagem AgentSpeak(L) em Python, visando eficiência computacional.

Agentes

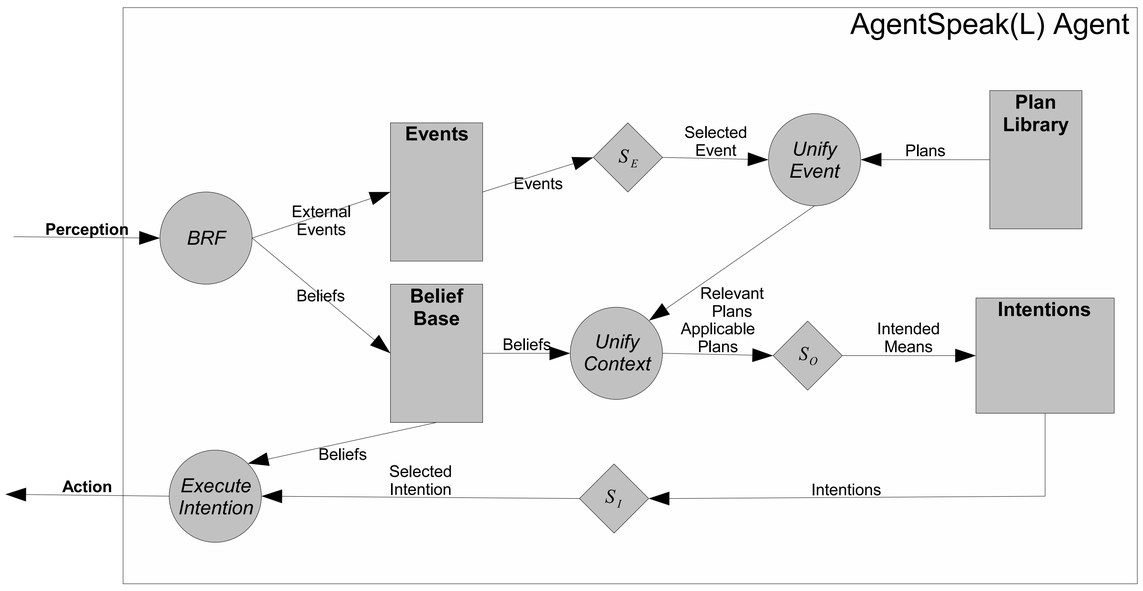
- Um agente é um sistema computacional que percebe o ambiente em que se encontra e realiza ações de forma autônoma para alcançar seus objetivos [3];
- Uma das abordagens mais comuns para a construção de agentes baseados em objetivos é utilizar estados mentais semelhantes ao raciocínio prático humano, como crenças, desejos e intenções (BDI).

AgentSpeak(L)

- O AgentSpeak(L) [2] é uma linguagem abstrata de programação orientada a agentes baseada na arquitetura BDI;
- Um programa desenvolvido em AgentSpeak(L) é especificado por um conjunto de crenças e planos. As crenças representam o conhecimento do agente sobre o seu ambiente e sobre si mesmo. Os planos são compostos por um cabeçalho, que especifica as circunstâncias iniciais para a sua execução, e um corpo que contém a sequência de ações que um agente irá realizar no ambiente;
- O código abaixo exemplifica um programa desenvolvido em AgentSpeak(L):

```
1    /* Objetivos Iniciais */
2    !start.
3    /* Planos */
4    +!start : true <- aloha; ?continue(execute);
   !run(poster).
5    +!run(A) : true <- mahalo(A).
```

- As definições do AgentSpeak(L) permitem elaborar uma sequência de passos que representa o raciocínio de um agente a cada ciclo:
  1. O agente recebe as informações dos estados do ambiente;
  2. A função de revisão de crenças (*BRF*) compara as percepções recebidas com B. As crenças diferentes são modificadas e geram um novo evento que é adicionado em E;
  3. S_E escolhe um único evento de E;
  4. A função *Unify Event* unifica o evento selecionado com os eventos ativadores, contidos no cabeçalho dos planos de P, para encontrar os planos relevantes;
  5. A função *Unify Context* unifica as variáveis do contexto dos planos relevantes com as crenças em B, para encontrar os planos aplicáveis;
  6. S_O escolhe um único plano pretendido que atualiza I;
  7. S_I seleciona a intenção contida no topo de I;
  8. A função *Execute Intention* executa a fórmula do corpo da intenção, gerando uma ação no ambiente;
- A figura a seguir ilustra este ciclo de raciocínio [1].



AgentSpeak(Py)

- O AgentSpeak(Py) é um interpretador da linguagem AgentSpeak(L) desenvolvido em Python;
- Foi concebido como um projeto *open source* e disponibilizado livremente para utilização pela comunidade;
- Neste trabalho, implementamos as funcionalidades abaixo:
  - Configuração do projeto através de um arquivo *.maspy*, que contém a lista dos agentes e a classe com a descrição do ambiente;
  - Execução do ciclo de raciocínio dos agentes descritos em AgentSpeak(L);
  - Descrição do comportamento do ambiente em Python a partir da extensão da classe `Environment`;
  - Execução das ações dos agentes no ambiente;
  - Função `.print()` para impressão de mensagens e do conjunto de crenças dos agentes;
  - Função `.send()` para troca de mensagens entre agentes, com o objetivo de atualizar crenças e adicionar ou remover objetivos;
  - Modo de depuração para visualização, a cada ciclo de interpretação, dos estados do ambiente, dos estados mentais dos agentes e de suas mensagens;

Experimentos e Resultados

- Neste trabalho, elaboramos 4 cenários em AgentSpeak(L) para demostrar a execução do interpretador;
- A tabela abaixo contém os resultados preliminares do tempo médio de execução ($\bar{t}$) e do desvio padrão (σ), ambos em segundos, para 2 destes cenários.

Cenário	***Número de Agentes | **Interações com o Ambiente**			
	500	**1000**	**2500**	**10000**
Poster*	$\bar{t} = 0.4871$ $\sigma = 0.0263$	$\bar{t} = 0.9759$ $\sigma = 0.0573$	$\bar{t} = 2.5642$ $\sigma = 0.0399$	$\bar{t} = 11.3012$ $\sigma = 0.0863$
Room**	$\bar{t} = 0.3645$ $\sigma = 0.0241$	$\bar{t} = 0.7044$ $\sigma = 0.0225$	$\bar{t} = 1.7252$ $\sigma = 0.0390$	$\bar{t} = 6.6474$ $\sigma = 0.1607$

Referências

[1] Felipe Meneguzzi. *Extending Agent Languages for Multiagent Domains*. PhD thesis, King's College London, 2009.

[2] Anand S. Rao. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1996.

[3] M Wooldridge. *Reasoning About Rational Agents*. MIT Press, 2000.