

Progetto IA:Graphlog

Davide De Zuane, Rahmi El Mechri

Idea

Analisi proprietà dei grafi.

Natura dell'Ambiente

Descrizione

- ▶ Ambiente: Teoria dei grafi
- ▶ Metriche di performance: nessuna
- ▶ Attuatori: interprete prolog
- ▶ Sensori: interfaccia utente

Proprietà



- ▶ Completamente Osservabile
- ▶ Agente singolo
- ▶ Deterministico
- ▶ Episodico
- ▶ Semi-statico
- ▶ Discreto

Agente

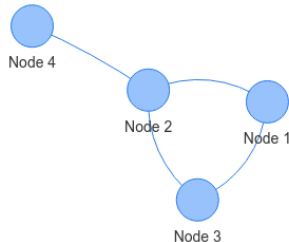
- ▶ Simple reflex
- ▶ Percezione: grafo caricato dall'utente
- ▶ Azione: calcolo principali caratteristiche del grafo

Tecnologie

Abbiamo realizzato una Single Page Application. Per realizzarla ci siamo serviti delle seguenti librerie:

- ▶  Tau-prolog: libreria javascript interprete prolog
- ▶  Vis.js: libreria javascript per la visualizzazione di grafi

```
function addNode(node) {  
  data.nodes.add(node);  
  //Add to Prolog KB  
  pl_kb_nodes_string += "node("+node.id+")."  
  session.consult(pl_kb_nodes_string+pl_kb_edges_string, {  
    success: function () {  
      console.log("Node added to KB successfully");  
      session.query("node(X).", {  
        success: function (goal) {  
          console.log("Query parsing went well");  
          session.answers(puri);  
        },  
        error: function (err) {  
          console.log("Query parsing went bad");  
        },  
      });  
    },  
    error: function (err) {  
      console.log("Node not added");  
    },  
  });  
}
```



Knowledge Base

La Knowledge Base è definita da due parti.

Fatti

I fatti vengono generati dinamicamente a partire dal grafo definito dall'utente, e sono definiti nel seguente modo: `+ node(x) .` `+ edge(x,y) .` `##` Regole Le regole vengono importate da un file prolog presente nel web server. Nel definire le regole abbiamo considerato solamente grafi simmetrici, tramite le regole: `+ edge_s(X,Y) :- edge(Y,X) .` `+ connected(X,Y) :- edge_s(X,Y) ; edge(X,Y) .`

Regole - 1

Regole - 2

Regole - 3

Javascript

GUI

Ringraziamenti

Grazie per l'attenzione!