

UNIVERSITÀ DEGLI STUDI DI VERONA

Intelligenza Artificiale

RIASSUNTO DEI PRINCIPALI ARGOMENTI

Davide Bianchi

March 22, 2018

Contents

1	Agenti razionali	2
2	Problemi di ricerca	2
2.1	Formulazione di problemi a stato singolo	3
2.2	Ricerca su alberi	3
2.3	Ricerca non informata	3

1 Agenti razionali

Agenti. Un agente è semplicemente un'entità che riceve percezioni e produce una risposta con delle azioni. Formalmente un agente è una funzione

$$f : P^* \rightarrow A$$

dove P^* è lo storico delle percezioni e A è un insieme di azioni.

Notare che se un agente ha $|P|$ possibili percezioni in ingresso, dopo T unità di tempo la funzione agente avrà il seguente numero di entries:

$$\sum_{t=1}^T |P|^t$$

Un agente è in generale una struttura formata da un'architettura fisica e un programma, e prende in input una percezione attuale e ritorna in output l'azione successiva da svolgere.

Esistono principalmente 4 tipi di agenti:

- agenti *simple-reflex*;
- agenti *reflex* con stato;
- agenti *goal-based*;
- agenti *utility-based*.

Performace measure. La *performance-measure* costituisce una sorta di punteggio che misura il comportamento dell'agente nell'ambiente in cui opera. Quindi, data una performance measure e le percezioni attuale dell'agente, questo sceglie la sequenze di azioni che la massimizzano.

Ambienti. Un ambiente, ovvero lo spazio in cui l'agente opera, è caratterizzato dai seguenti tratti:

- Osservabilità;
- Determinismo;
- Episodicità;
- Staticità;
- Discretezza;
- Presenza di altri agenti.

Il tipo di ambiente ovviamente condiziona il design degli agenti che vi operano.

2 Problemi di ricerca

Dividiamo i problemi in due macro-categorie:

- Deterministici e completamente osservabili, richiedono un singolo stato;
- Non osservabili, in tal caso gli agenti non sanno dove la soluzione possa risiedere;
- Non deterministici o parzialmente osservabili; problema di contingenza/eventualità (??);
- Lo spazio degli stati è sconosciuto (problemi di esplorazione).

2.1 Formulazione di problemi a stato singolo

Un problema a stato singolo è definito da 4 elementi:

1. uno stato iniziale;
2. una funzione successore (insieme di coppie azione-stato successivo);
3. un test di *goal*;
4. costo del percorso (costo dei singoli step).

Una soluzione è quindi una sequenza di azioni che portano dallo stato iniziale allo stato di *goal*.

2.2 Ricerca su alberi

In generale ogni algoritmo di ricerca su alberi funziona come segue:

```
function tree-search(problem, strategy)
  inizializza l'albero di ricerca usando lo stato iniziale del problema;
  loop:
    se non ci sono candidati per l'espansione:
      return failure
    scegli un nodo foglia per l'espansione rispettando strategy;
    se il nodo contiene uno stato goal:
      return solution;
    altrimenti:
      espandi il nodo e aggiungi il nodo risultante all'albero
```

Nota: un nodo è una struttura dati, uno stato è una rappresentazione fisica di un nodo, non ha stati padre, ecc.

Il metodo generale per eseguire una ricerca sugli alberi è il seguente:

```
function tree-search(problem, fringe):
  fringe = insert(make-node(initial-state[problem]), fringe)
  loop:
    if fringe is empty
      return failure
    if goal-test(problem, state(node))
      return node
    fringe = insert-all(expand(node, problem), fringe)
```

2.3 Ricerca non informata

I problemi di ricerca non informata utilizzano solo le informazioni presenti nella definizione del problema.

Breadth-first search. Espande il nodo non espanso più in superficie.