

TeamDB Agent

Componenti: Diego Buratti

I due agenti realizzati dal team vengono implementati dalle classi `AgenteDB.py` e `AgenteDBNoLimit.py`

Agente MultiVAC (`AgenteDB.py`)

Questo agente sfrutta come memoria 3 liste di 7 elementi:

- `visited`: nodi già visitati
- `wall`: muri incontrati
- `ag2visited`: nodi visitati dagli altri agenti

Per realizzare la limitazione di memoria a solo 7 elementi vengono sfruttati tre contatori, uno per vettore che vengono re-inizializzati a 0 raggiunto il limite di 7.

L'agente memorizza inoltre la sua attuale posizione (*coord*) e l'ultima operazione effettuata(*lastpos*).

La posizione dei vicini viene ottenuta sommando la posizione attuale dell'agente con le posizioni relative degli altri agenti ottenuta dalla scansione del vettore *neighbors* dato in input.

La classe nel suo metodo principale inserisce la posizione degli altri agenti nel vettore *ag2visited* controlla lo stato attuale dell'agente; se lo stato indica *Dirty* restituisce il comando di pulizia. Nel caso in cui lo stato si *Bump* l'agente aggiunge la coordinata del muro alla lista *wall* aggiornando il conteggio relativo al vettore dei muri e riportando la posizione attuale alla posizione precedente. Viene inoltre decremento il contatore che controlla le posizioni visitate per evitare la ripetizione della posizione in entrambe le liste.

Il cuore della classe è nella funzione *nextDir()* che viene richiamata nel momento in cui lo stato è diverso dai due controllati precedentemente. In questo caso si estrae a caso un valore tra 0 e 3 che rappresenta la direzione da prendere (0='GoEast', 1='GoWest', 2='GoNorth', 3='GoSouth'). A questo punto viene controllato se la posizione raggiungibile dalla direzione estratta sia raggiungibile secondo queste condizioni:

1. la posizione non sia già stata visitata
2. la posizione non sia un muro già urtato
3. la posizione non sia stata raggiunta da un vicino

Questa ultima condizione garantisce che gli agenti lavorino in maniera **collaborativa** non sovrapponendo le pulizie delle caselle. Essendo la memoria limitata in questo agente questo è vero per le ultime 7 posizioni visitate dal vicino.

Se queste condizioni più stringenti non vengono soddisfatte la funzione permettere di scegliere una altra direzione sempre estratta nuovamente a caso senza il vincolo delle posizione già visitate dall'agente.

Scelta la direzione viene aggiornata la posizione, aggiunta ai nodi visitati e restituita per lo spostamento.

Viene inoltre impostato un limite di spostamenti senza pulizia per l'arresto dell'agente (*NoOp*).

Agente MultiVAF (AgenteDBNoLimit.py)

La versione dell'agente senza limitazione di memoria si comporta in maniera del tutto identica al precedente con la differenza che le liste utilizzate per il controllo non hanno una dimensione prefissata ma illimitata, quindi non sono più necessari contatori di controllo. I due agenti non essendoci limitazioni di memoria conosceranno ed eviteranno tutti i muri già incontrati, e le caselle visitate dai vicini senza più il limite dei 7 elementi.