Trabalho
Sistemas Multiagente

Francisco Vicenzi
Gustavo Kundlatsch

19 de outubro de 2020

Implementação

Colaboração

A colaboração entre os robôs mineradores foi implementada utilizando a comunicação, por meio do comando de broadcast.

O trecho de código 1 apresenta dois procedimentos que fazem parte da checagem por recursos: *check_for_resources*. A primeira parte faz com que o agente avise a todos agentes a posição de um recurso que está sendo requisitado pelo agente chefe, através de *broadcast(tell, resource(X,Y,R))*. O segundo, avisa da posição de recursos que serão buscados mais tarde, já que o recurso necessário, R, não é o mesmo que foi achado, S.

```
+!check_for_resources
   :  resource_needed(R) & found(R) & my_pos(X,Y)
   <- !stop_checking;
      .broadcast(tell,resource(X,Y,R));
      !take(R,boss);
      !continue_mine.

+!check_for_resources
   :  resource_needed(R) & found(S) & my_pos(X,Y)
   <- .broadcast(tell,resource(X,Y,S));
      move_to(next_cell).
```

Listing 1: Utilização do broadcast com *tell*

O trecho 2 mostra o passo de remoção da crença de recursos que não precisam mais serem coletados, a partir de *broadcast(untell, resource(X,Y,R))*.

A crença *resource(X,Y,R)*, utilizado em ambos os trechos, foi adicionada ao agente para trocar a informação de posição dos recursos entre os agentes. X e Y guardam a posição do recurso, e R seu tipo.

```
+!check_for_resources
   :  resource_needed(R) & not found(R) & my_pos(X,Y)
   <- move_to(next_cell);
      .broadcast(untell,resource(X,Y,R)).
```

Listing 2: Utilização do broadcast com *untell*

Movimentação

As regras descritas em 3 tratam a movimentação do agente em relação as crenças adicionadas pelo *broadcast*. O primeiro procedimento faz com que o agente, ao receber uma nova crença de *resource*, verifique se o recurso é do tipo requisitado pelo chefe. Caso isso seja verdade, o agente se movimentará para a posição indicada. O segundo procedimento é acionado quando o chefe troca o tipo de recurso requisitado. Desse modo, o agente é dirigido a um recurso já encontrado previamente, do tipo solicitado. Por fim, o terceiro procedimento lida com desejo de movimentação do agente, que é adequado à crença em *resource*.

```
+resource(X,Y,R)
   : not my_pos(X,Y) & resource_needed(R)
   <- !go(X,Y).

+resource_needed(R)
   : resource(X,Y,R)
   <- !go(X,Y).

+!go(X,Y) : true
   <- move_towards(X,Y).
```

Listing 3: Regras de movimentação.

Caminhos

O desafio, que era fazer com que os agentes evitem caminhos repetidos, não foi implementado.