

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO
INTELIGÊNCIA ARTIFICIAL**

Sistemas Multiagente

LUCAS CASTRO TRUPPEL MACHADO

FLORIANÓPOLIS, 2 DE JUNHO DE 2024

1)

Para implementar a colaboração entre agentes, foi criada uma versão do plano `+!notify_resource_found` que notifica os outros robôs sobre a posição de um recurso encontrado. Isso ocorre apenas se o recurso já não tiver sido registrado como encontrado, ou seja, se não existe a crença `resource_found(R,X,Y)`.

```
14 // Se encontrou um recurso necessário que ainda não foi encontrado
15 // Então notifica os outros agentes, pega o recurso e continua a mineração
16 +!check_for_resources
17 : resource_needed(R) & found(R) & my_pos(X,Y) & not resource_found(R,X,Y)
18 <- !stop_checking;
19     !notify_resource_found(R,X,Y);
20     !take(R,boss);
21     !continue_mine.
```

```
79 +!notify_resource_found(R,X,Y) : true
80 <- +resource_found(R,X,Y);
81     .print("broadcast resource found R=", R, " X=", X, " Y=", Y);
82     .broadcast(tell, resource_found(R,X,Y)).
83
```

Assim, os outros robôs terão a crença da existência do recurso nessa posição e irão até ela caso estejam vagos. Se os outros robôs já estiverem minerando, a crença será adicionada, mas continuarão minerando o recurso anterior.

```
68 // Se não encontrou nenhum recurso e conhece um recurso desejado que ainda não foi esgotado
69 // Então vai para a posição do recurso desejado
70 +!check_for_resources
71 : not found(_) & resource_needed(R) & resource_found(R,X,Y) & not resource_finished(R,X,Y)
72 <- -checking_cells;
73     ?my_pos(Xo,Yo);
74     +prev_pos(Xo,Yo);
75     !go_cord(X,Y);
76     +checking_cells;
77     !check_for_resources.
```

2)

Quando um agente não encontra o recurso na posição que acreditava possuir o recurso, mas ainda não existe o conhecimento do esgotamento do recurso naquela posição, pela crença `resource_finihsed(R,X, Y)`, ele realiza um broadcast indicando o esgotamento do recurso naquela posição. Dessa forma, os agentes não irão mais buscar o recurso nessa posição.

```
51 // Se chegou na posição de um recurso necessário que já foi encontrado, foi esgotado, mas não foi marcado como esgotado
52 // Então notifica os outros agentes sobre o esgotamento e move para a próxima célula
53 +!check_for_resources
54 : resource_needed(R) & not found(R) & my_pos(X,Y) & resource_found(R,X,Y) & not resource_finished(R,X,Y)
55 <- !notify_resource_finihsed(R,X,Y);
56     !move_somewhere.
```

```
84 +!notify_resource_finihsed(R,X,Y) : true
85 <- +resource_finished(R,X,Y);
86     .print("broadcast resource finished R=", R, " X=", X, " Y=", Y);
87     .broadcast(tell, resource_finished(R,X,Y)).
```

3)

Ao encontrar um recurso ainda não procurado, mas que será necessário futuramente, o agente realizará um broadcast indicando a existência do recurso na posição, com a crença `resource_found(R,X,Y)`.

Quando o recurso sendo procurado for esse, os robôs já conterão esse conhecimento e o se deslocarão para sua posição, da mesma forma que acontece anteriormente.

```
50
51 // Se chegou na posição de um recurso necessário que já foi encontrado, foi esgotado, mas não foi marcado como esgotado
52 // Então notifica os outros agentes sobre o esgotamento e move para a próxima célula
53 +!check_for_resources
54 : resource_needed(R) & not found(R) & my_pos(X,Y) & resource_found(R,X,Y) & not resource_finished(R,X,Y)
55 <- !notify_resource_finished(R,X,Y);
56 !move_somewhere.
```

4)

Antes de se deslocar para a posição que foi descoberta pela mensagem recebida, a posição anterior é guardada na crença `prev_pos`.

```
68 // Se não encontrou nenhum recurso e conhece um recurso desejado que ainda não foi esgotado
69 // Então vai para a posição do recurso desejado
70 +!check_for_resources
71 : not found(_) & resource_needed(R) & resource_found(R,X,Y) & not resource_finished(R,X,Y)
72 <- -checking_cells;
73 ?my_pos(Xo,Yo);
74 +prev_pos(Xo,Yo);
75 !go_cord(X,Y);
76 +checking_cells;
77 !check_for_resources.
```

Essa informação é utilizada para retornar à posição original ao terminar de minerar o recurso da posição notificada.

```
31 // Retorna para posição antes ser chamado para ajudar na coleta
32 +!move_somewhere : prev_pos(X1,Y1)
33 <- -checking_cells;
34 .print("move to prev_pos X=", X1, " Y=", Y1);
35 !go_cord(X1,Y1);
36 -prev_pos(X1,Y1);
37 +checking_cells;
38 !check_for_resources.
```