

担当 山影進

TA 阪本拓人、鈴木一敏、保城広至、
光辻克馬、山本和也

第四回 エージェント判断する！(5月17日)

概略

前回の課題(回転、酔っぱらい、スケーター)

変数の種類

変数の型

if文のいろいろ

課題

前回の課題(回転、酔っぱらい、スケーター)

7名の方に提出していただきました。

変数の種類(ツリーの中の変数、ルールの中の変数)

変数には、大きく2種類のものがあります。一つは、ツリーに表示される「正式な」変数です。前回、作成した「Speed」という変数はこれに該当します。もう一つはルールの中で定義される「一時的な」変数です。これは、ルールを記述するうえで必要に応じて作成される変数で、ツリーには表示させずに用います。

正式な変数：

ツリーの中で定義します。それぞれのエージェントが、それぞれ異なった値を持ち歩かなければならないようなタイプの変数にはこちらの変数を定義してやります。

一時的な変数：

ルールの中で「dim(変数名) as(変数の型)」の形で定義します。変数をルールの中で使ったあと、その値を保存する必要のないようなタイプの変数にはこちらの変数を定義してやります。

両者の最大の違いは、「正式な変数」はそれぞれのエージェント(例えばそれぞれ赤亀)が、その変数のそれぞれの値を恒常的に備えるのに対し、「一時的な変数」はそのとき処理して

いるエージェントについてのみ、その変数の値が設定され、処理の順番が次のエージェントにうつった時点で、その値は初期化されてしまうという点にあります。それぞれに特徴があるので、うまく使い分ける必要があります。書いたルールは、たくさんいる全てのエージェントに適用されるものであることを頭に置いておいて、値の保存の必要があれば、ツリーの中に変数を設定し、不要であればルールの中で定義してやってください。

変数の型

変数は、「型」を必ず持っている（変数を作る際に設定させられましたね）、あらかじめどいうタイプの値をとるのかを限定しておく必要があります。最も、イメージしやすく、一般的なものは、「整数型」や「実数型」ですが、以下のようなさまざまな型の変数を設定することができます。

型の種類	正式名称	中に入るもの	具体的な値
ブール型	Boolean	真偽	Ture, False
文字列型	String	文字列	Suzuki, Yamamoto
整数型	Integer	整数	365, 1,2,3
実数型	Double	実数	3.1415..., 1.1415...
空間型	Space	空間名	SpaceK, 空間
エージェント種別型	AgtType	エージェントの種類	赤亀、青亀
エージェント型	Agt	エージェント	赤亀01、赤亀02
エージェントセット型	AgtSet	エージェントの集合	{赤亀01、赤亀02}、{赤亀00、青亀01}

今日は上の4つについて、把握しておいてください。

とりうる値の限界については、ヘルプを参照にしてください。

ツリーの中の変数においても、ルールの中の変数においても型は、必ず必要です。

if 文

基本の if 文 「もし～～だったら、 しなさい」

```

if [ 条件式～～ ] then
    [ 実行したいルール ]
end if

```

基本的な if 文の書き方です。エージェントにさまざまな判断をさせるときに用います。例えば、お金のない人とは友達にならないエージェントの行動は、以下のようなルールをステップ (Agt_Step) に書くことで表現できます。

```
if <友達の所持金 == 0> then
    <つきあわない>
end if
```

==の書き方に注意してください。前回学んだように、=は代入を表すので、同値であることを表す際に、==を用います。(とてもよく間違えます)

if else 文 「もし～～だったら、<1>しなさい。
 そうでなかったら、<2>しなさい。」

```
if [条件式～～] then
    [実行したいルール<1>]
else
    [実行したいルール<2>]
end if
```

elseif 文 「もし～1～だったら、<1>しなさい。
 そうでなくもし～2～だったら、<2>しなさい。
 もしそうでなかったら、<3>しなさい。」

```
if [条件式～1～]                      then
    [実行したいルール<1>]
elseif [条件式～2～]                      then
    [実行したいルール<2>]
else
    [実行したいルール<3>]
end if
```

課題

山影研究室の HP から < 課題初期モデル > をダウンロードして、以下のようなルールを書いてみましょう。

初期モデルは、100*100 の空間があり、その真ん中から方向とスピードをランダムに(最高 5、最低 0) 決めた歩行者が、ばらばらに歩き始めるモデルになっています。

これらの課題には、正解があるわけではありません。どう表現するかは、モデル作成者の自由(センス?力量?)です。

- (1) 早すぎる歩行者は、徐々にスピードを緩め、遅すぎる歩行者は徐々にスピードを速め、やがてみんなのスピードが「3」にそろろう。
- (2) スピードの速いものは右周りをし、遅い者は左回りをする。
- (3) 10%の確率で、自分の方向とスピードをランダムに設定し直す。

- (4) 空間の端を壁にそって回り続ける歩行者。
- (5) 進むべき方角が決まっていて(例えば 45°) 最初にばらばらに歩き始めるが、徐々に進むべき方角に方向を変えていく歩行者。
- (6) 進むべき方角がそれぞれに決まっていて(これもランダムに決定する) 最初はばらばらに歩き始めるが、徐々に「自分の」進むべき方角に方向を変えていく歩行者。
- (7) 歩き出して、空間の 4 スミのうち、一番近いスミに向かう歩行者。
- (8) 空間の端に来たら方向を転換して、ループしないようにする歩行者。