

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

第六回 神様登場！(5月31日)

概略

セル型のモデル

コントロールパネルの設定

For 文の説明

CreateAgt の説明

RandomPutAgtSetCell の説明

MakeAgtSet の説明

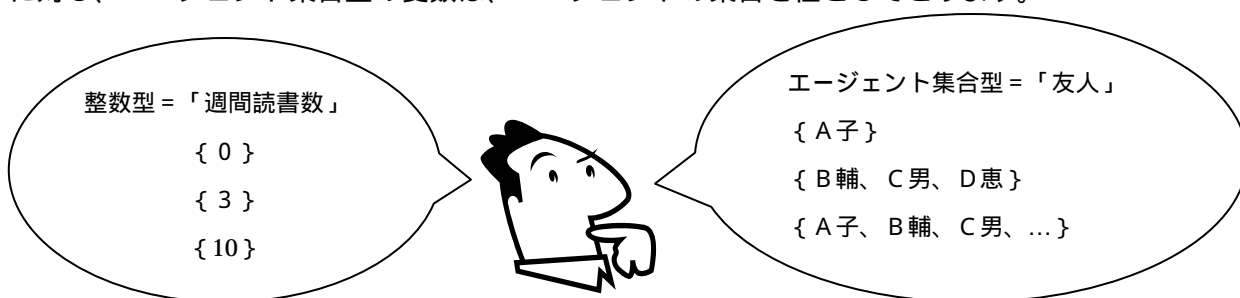
MergeAgtSet の説明

課題

前回学んだこと

エージェント集合型変数

エージェントの集合を格納できる変数です。整数型の変数が、1 や 2 といった値をとるのに対し、エージェント集合型の変数は、エージェントの集合を値としてとります。



エージェント集合型関数

KK-MAS には、エージェント集合型変数をあつかうためのルール(関数)が多数、用意されています。そのうちの2つについて学びました。

- **MakeOneAgtSetAroundOwn**(Agt 集合型変数, 範囲, Agt 種別, 自分自身を含む?)
自分の周りのエージェントのリストをエージェント集合型変数の中に格納する。
- **CountAgtSet**(エージェント集合型変数)
エージェント集合に格納されているエージェントの数を数える。

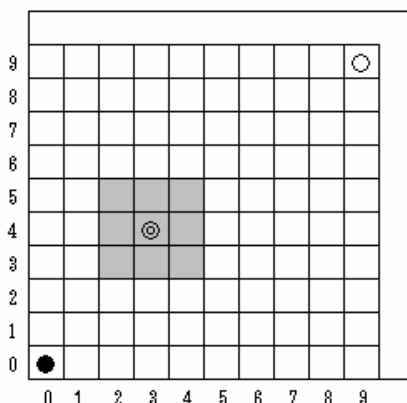
今回のねらい

マルチエージェント型のシミュレーションは、エージェントに行動のルールを与えてやり、エージェント同士を相互作用させることが中心となります。しかし、それを補助するために、Universe というもの（神様、モデル作成者の代理人）が存在します。Universe に変数やルールを設定してやることで、モデル全体に関わることを処理させたり、モデル全体の状態を集計させたりすることができます。今回と次回は、分居モデルを作成しつつ、Universe で変数やルールを処理する方法を学びます。

セル型のモデル

これまで三回は、空間の中をエージェントが自由に動き回るモデルを扱いましたが、今回からは、空間にマス状の格子があってそのマスに入る形でエージェントが存在する（移動する）モデルについて学びます。将棋やオセロを思い浮かべてください。こういうタイプの空間表現をセル型と呼びます。

- (1) 座標は整数のみ。
- (2) さらに、座標は 0 から数える。
- (3) 近傍の形が違う。
- (4) 出力範囲を修正してね。



前回までの Forward(自分の Direction の方向に進む)に代わり、セル型空間の中では、ForwardXCell と ForwardYCell というルールを用います。

ForwardXCell(整数) = X 軸方向に マス進む

ForwardYCell(整数) = Y 軸方向に マス進む

では、分居モデルの舞台となる街を作ってみましょう。基本的な作り方は、今までと全く同じです。大きさは 10*10 にします。

エージェント（赤亀、青亀）もついでに作っておきましょう。

Universe : ルールの実行順序

Universe のルールエディタにルールを書き込むことで、いろいろなことを実行させることができます。その実行順序について説明しておきます。エディタを開けてみてください。Universe のルールエディタは、Univ_Init{}、Univ_Step_Begin{}、Univ_Step_End{}、Univ_Finish{}の4つの部分に分かれています。

Univ_Init{}	シミュレーションの最初に一度だけ実行される
Univ_Step_Begin{}	各ステップの初めに実行される
Univ_Step_End{}	各ステップの終わりに実行される
Univ_Finish{}	シミュレーションが終了するときに一度だけ実行される

コントロールパネルを作る

モデルにコントロールパネルを設定することで、動かしているモデルの設定を外からいじることができるようにすることが出来ます。コントロールパネルに設定することができるのは、Universe レベルの変数のみです。

「赤亀の数」「青亀の数」を表す変数を Universe に設定し、それをコントロールパネルで設定できるようにしてみましょう。(まだルールも何も書いていないので、ここでは設定だけです) [設定 > コントロールパネル設定 > 追加] で設定します。

今日の文法

For 文 (繰り返し文)

If 文 (条件文) と並ぶ基本命令文です。指定した回数だけ、命令を繰り返させるのに用います。以下のように書けば、ルール A を 10 回繰り返します。

```
dim i as integer
For i = 0 to 9
    <ルールA>
Next i
```

エージェントの ID 番号 (ちょっと上級)

少し難しいですが、**For** 文と組み合わせることで用いることが出来るので、エージェントの ID 番号についてここで説明します。赤亀、青亀といったエージェントは、ツリーの中には一つしか書かれていませんが、実はたくさん (エージェント数) だけ存在します。それぞれのエージェントを区別したいときには、エージェントの ID 番号というものを uses。

たとえば、10 匹の赤亀が存在するときには、RedTurtle(0), RedTurtle(1), RedTurtle(2)..., RedTurtle(9) が存在します。10 匹の赤亀の中のある一匹の赤亀 (例えば背番号 5) の X 座標を操作したいときには、以下のように書くことで操作できます。

Ex. Universe.Streets.RedTurtle.X(5) = 9

この ID 番号と For 文を組み合わせることで、Universe がエージェントを (ある程度) 規則的に配置させたり、操作したりすることを表現することもできます。

```
dim i as integer
For i = 0 to 9
    Universe.Streets.RedTurtle.X(i) = i
    Universe.Streets.RedTurtle.Y(i) = i
Next i
```

CreateAgt エージェントを創り出す

指定した種類のエージェントを一人創り出します。1つの条件 (= 引数) を設定してやります。

```
CreateAgt( 創り出すエージェント種別 )
```

Ex. CreateAgt(Universe.Streets.RedTurtle)

ここまでのルールで、コントロールパネルで指定した数の赤亀、青亀を、Universe が最初に創り出すルールを書いてみましょう。

以下では、エージェント集合型関数をさらに幾つか勉強します。

RandomPutAgtSetCell 集合に含まれるエージェントをランダムに配置する
指定したエージェント集合に含まれているエージェントを空間内にランダムに配置する関
数です。セル型に配置するために、Cell はセル型に配置することを意味します。

```
RandomPutAgtSetCell( エージェント集合型変数, 重なりを許すか ( True/False ) )
```

Ex. RandomPutAgtSetCell(Universe.Turtles, False) > ツリーで設定するとき

Ex. dim Turtles as AgtSet

```
RandomPutAgtSetCell(Turtles, False) > ルールで設定するとき  
(まえてルールの中で (先頭で) 宣言しておく必要があります)
```

MakeAgtSet

指定された種類のエージェント全部のリストをエージェント集合型変数に格納する
先週は、MakeOneAgtSetAroundOwnというエージェント集合型関数を学びました。自分
自身の周りがあるエージェントのリストをエージェント集合型変数の中に格納する変数で
した。このようにさまざまな条件をつけずに、ある種類のエージェントを全部、変数の中
に格納してしまう関数が、MakeAgtSetです。

```
MakeAgtSet( 生成されるエージェント集合型変数, エージェント種別 )
```

Ex. MakeAgtSet(Universe.Turtles, Universe.Streets.RedTurtle) > ツリー設定

Ex. dim Turtles As AgtSet

```
MakeAgtSet(Turtles, Universe.Streets.RedTurtle) > ルール設定
```

MergeAgtSet エージェント集合を合体させる

あるエージェント集合型変数に、他のエージェント集合型変数に含まれているエージェン
トのリストを加えてしまう関数です。2つの条件を設定してやります。変数2に含まれて
いるエージェントのリストを変数1に格納することができます。

```
MergeAgtSet( エージェント集合型変数 1, エージェント集合型変数 2 )
```

Ex. MergeAgtSet(Universe.Turtles1, Universe.Turtles2) > ツリー設定

Ex. dim Turtles1 As AgtSet

```
dim Turtles2 As AgtSet
```

```
MergeAgtSet(Turtles1, Turtles2) > ルール設定
```

以上のルールで、創り出された赤亀と青亀を空間の中にランダムにばらまくルールを書いてみましょう。

これで、分居モデルのベースとなるモデルができました。

課題

<課題初期モデル>には、 10×10 のセル型の空間と10匹の亀が用意されています。

- (1) 左端に10匹を整列させてみましょう。
- (2) 斜めに10匹を整列させてみましょう。
- (3) 10匹を行進させてみましょう。
- (4) 右端、左端に亀が来たら、折り返すルールを書いてみましょう。
- (5) 最初に、亀がランダムに配置されるようにします。
- (6) 亀が気まぐれに、縦に動いたり横に動いたり、右に行ったり左に行ったりするように設定します。
- (7) 自分の周囲に亀が3匹以上いたら、そこで定住するルールを書いてみましょう。
[MakeOneAgtSetAroundOwnCell(Agt 集合型変数、視野、Agt 種別、含不含)]