

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

第五回 エージェント周りを見る！(5月24日)

概略

前回の課題

変数の型 (エージェント型、エージェント集合型)

エージェント集合型関数

課題

前回の課題

4名の方に提出していただきました。

前回の復習

変数の種類

変数には、ツリーに表示される「正式な」変数と、ルールで設定される「一時的な」変数があります。> とりあえず心に留めておいてください。

変数の型

変数は常に型を持ちます。どんなタイプの値をとるのかをあらかじめ設定するものです。

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}

条件文

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
    
```

初期モデルの説明と今回の作業

今回、HP でダウンロードしてもらった初期モデルは、家路にいそぐ歩行者が描かれています。歩行者 A は東の方角 (Direction0) に家がある人々です。歩行者 B は西の方角 (Direction180) に家がある人々です。それぞれの歩行者は、帰る方角 (Destination) を持っていて、もし異なる方向に向かうことがあっても家の方角へと方向を変えていきます。

前回まで作成したエージェントは、自分で判断していましたが、その判断は一人よがりなもので、周りの世界と相互作用はしていませんでした。今回は、幾つか新しい文法事項を学んで、周りの世界を認識する方法を学びます。

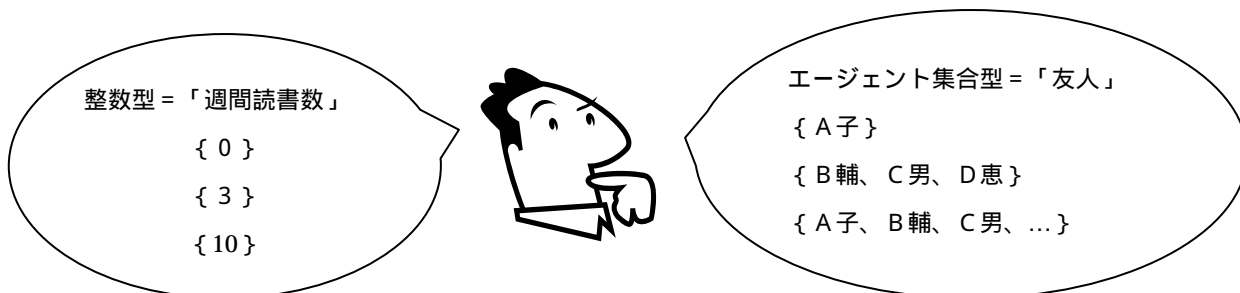
具体的には、近くにこちらに向かってくる歩行者がいるとぶつからないようによけようとする歩行者のモデルを作成します。

エージェント集合型

前回、変数の型について学びました。数字を格納できる整数型と実数型。文字列を格納できる文字列型などがありました。今回はエージェント集合(セット)型の変数を学びます。

エージェント集合(セット)型変数

エージェントの集合を格納できる変数です。少しわかりにくいですが、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。



> ツリーで定義するときは、「エージェント集合型」を選択してください。

(> ルール内で定義するときは、Dim <変数名> As AgtSet としてください)

では、歩行者 A に「近接者 (Neighbors)」という名称のエージェント集合型の変数を設定してあげてください。

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。（ヘルプ参照）それぞれ便利なものですが、今回は、その中から 2 つをお教えします。

MakeOneAgtSetAroundOwn 自分の周りを認識する

自分自身の座標の周辺にいる（視野範囲内の）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関数です。以下のように書いて用います。4 つの条件を設定してやります。（= これのことを引数といいます）

```
MakeOneAgtSetAroundOwn( エージェント集合型変数, 範囲, エージェント種別,  
自分自身を含むか含まないか )
```

Ex. MakeOneAgtSetAroundOwn(My.Neighbors, 1, Universe.Space_K.Walker, False)

では、さきほど設定したエージェント集合型変数（近接者（Neighbors））に、向かってくる歩行者（A にとっては B）のうち、距離 1 以内に近づいたものを格納するルールを書いてください。

CountAgtSet 数える

エージェント集合型変数の中に格納されているエージェントの数を数える関数です。以下のように書いて用います。1 つの条件を設定してやります。

```
CountAgtSet( エージェント集合型変数 )
```

Ex. My.Num_Neighbors = CountAgtSet(My.Neighbors)

では、歩行者たちが自分の近接者の数を数えるルールを書いてみましょう。確認してみましょう。（情報出力の方法）

モデルを完成させましょう。

(1) 近接者が一人でもいれば、右に 20 度、曲がるようにルールを書いてあげましょう。

(2) 西に向かう歩行者 B についても同じルールを書いてあげましょう。

() これでモデルは完成です。どんなことが起こるでしょう。

() 局所的相互作用 (bottom-up)

() エージェントの意図 (micro-motives) とシステムのふるまい (macro-behavior)

(3) 歩行者たちが 10 度しか曲がらなかったら、どんなことが起こるでしょう。

(4) 西に向かう歩行者だけが 10 度しか曲がらなかったら、どんなことが起こるでしょう。