

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 11 回 状態・属性を持たせる (7 月 5 日)

今日の目的:

今回は、文字列型変数を使って、エージェントに文化や思想、遺伝子など、より細かな状態や属性を持たせる方法を学びます。同時に、文字列の操作の方法、文字列を使った表示などを学びます。

文字列型を使って状態や属性を表す

文字列型変数とは、数字やアルファベットや漢字など色々なものを見たま文字列として格納する変数です。遺伝子、文化、思想などを表す時に便利です。(詳しくは後述) ツリーで作成するときにはプロパティで「文字列型」を指定します。ルール内で一時的に使用する場合は `dim s as string` などと宣言します。打った文字を文字列型変数に格納するときは、“ ”でくくります。たとえば、以下のように書きます。

```
s = "AGGCTGCTATCCTATCGAA"
```

```
s = "moji retu ha benri"
```

```
s = "0336042000"
```

```
s = "こんにちは。私リカちゃん。リカのお話聞いてくれる?"
```

これらは「見たままの文字」なので、たとえ数字であっても数字としての意味はありません。たとえば、整数型で 0001 は 1 と同じですが、文字列型の 0001 はあくまで 0001 です。したがって、文字列型の変数はくっつけることは出来ても、足し算できません。もし文字列の数字を計算に利用したい場合は、`cint()`、`cdbl()`などを用いて、整数型や実数型に変換してから行ってください。

整数型の扱い

```
i = 0 + 1 + 2
```

```
i == 3
```

文字列型の扱い

```
s = "0" & "1" & "2"
```

```
s == "012"
```

```
cint(s) == 12
```

```
cdbl(s) == 12.0
```

また、一部分を取り出したいときには、関数を用います。

```

s = "0336042000"
sigai = left(s, 2)           左から二文字取り出されます。
kyoku = mid(s, 3, 4)       3文字目から4文字取り出します。
matu = right(s, 4)        右から4文字取り出します。

```

なお、漢字やひらがななど全角の文字は、1文字で2文字と数えます。“こんにちは”であれば10文字です。文字列が同じかどうか確かめたいときには、StrComp()を使います。

```

if StrComp(s, t) == 0 then    s と t が同じであれば・・・(式の値は 0)
if StrComp(s, "あ") <> 0 then  s が「あ」でなければ・・・(式の値は 1 か-1)

```

そのほかにも、文字列の長さを返す関数や、特定の文字列を探す関数があります。文字列の長さが変わるモデルや、類似したパターンがあるか探すモデルなどで便利でしょう。(多少技術的になるので、課題の上級編で扱っています。使い方の例はその解答例を見てみてください。)

例題:

文化が伝播するモデルを作り、文化がどのように画一化するかを見ます。今回は文化を3桁の数字の文字列で表しましょう。それぞれの桁が特定の項目を表していると考えましょう。たとえば、建築様式であれば、各桁が、柱の形、外壁の素材、屋根の形状などを表していると考えると良いでしょう。エージェントは毎ステップ、周囲1近傍の他人のうち一人から影響を受けて、1つの要素だけを真似します。文化が違くと違う色になるようにします。

- ・ 10×10のループした空間と、エージェントを100個作ってください。エージェント直下には、文化を表す文字列型変数と、色を表す整数型変数を作っておきましょう。
- ・ エージェントを配置し、初期の文化を持たせます。

```

Agt_Init{
my.X = my.id mod 10          id ÷ 10 のあまり(0~9)。空間の幅が 10。
my.Y = my.id \ 10           id ÷ 10 の商(0~9)。空間の高さが 10。
                             (JAVA 版 KK-MAS では「¥」を打つと「\」になります。)
my.cultura = cstr(cint(rnd()*10)) & cstr(cint(rnd()*10)) & cstr(cint(rnd()*10))
                             0 以上 10 未満の乱数の小数点以下を切り捨てて実数にして、それを文字列型に変更します。それを 3
                             回行い、&でつなげています。000 から 999 までの数字が文字列として格納されます。
}

```

- ・ とりあえず、固定色で出力設定してみましよう。変数情報表示で文化を表示してみましよう。ここまで出来たら、毎ステップの行動ルールを書いていきます。

```

Agt_Step{
dim mawari as agtset
dim kazu as integer
dim aite as agt
dim r as double

makeallagtsetarounddown(mawari, 1, false)    自分を除く周囲 1 近傍の集合を作ります。
kazu = countagtset(mawari)                  その中に含まれるエージェントを数えます。
aite = getagt(mawari, cint(kazu*rnd()))      そのうち一人をランダムに選び aite に格納。

if StrComp(my.cultura, aite.cultura) <> 0 then    もし aite の文化が自分と違ったら・・・
  r = rnd()*3                                    1 以上 3 未満のランダムな数を作ります。
  if r < 1 then                                  1 未満だったら、
    my.cultura = mid(aite.cultura, 1, 1) & mid(my.cultura, 2, 1) & mid(my.cultura, 3, 1)
    1 桁目だけ aite の 1 桁目と入れ替え、それ以外は自分の文化を保持。
  elseif r < 2 then                              そうでなくて 2 未満だったら、
    my.cultura = mid(my.cultura, 1, 1) & mid(aite.cultura, 2, 1) & mid(my.cultura, 3, 1)
    2 桁目だけ変える
  else                                            それ以外(2 以上 3 未満)だったら
    my.cultura = mid(my.cultura, 1, 1) & mid(my.cultura, 2, 1) & mid(aite.cultura, 3, 1)
    3 桁目だけ変える
  end if
end if

//RGB()で、1 桁目×25、2 桁目×25 という風にして色を付ける。
my.iro = RGB(cint(mid(my.cultura, 1, 1))*25, cint(mid(my.cultura, 2, 1))*25, cint(mid(my.cultura, 3,
1))*25)
}

```

課題

上記の例題の mid の部分を、left()、right()を使って書き直してみましょう。

Universe にそのステップにあった変化の総数を表す変数を作り、変化が無くなったら、「終了しました。」というメッセージが出るようにしてみましょう。文字列を表示する方法はいくつもあります。試してみましょう。

- 毎ステップ終わりに print() でコンソール画面に出力メッセージを出力。
- exitsimulationmsg() の引数にメッセージを指定。(表示が出て試行が終了します。)
- Universe に文字列型変数を作り数値画面出力 その中に文字列を格納

***すこし上級(この位まではできるでしょう。)**

今回学んだタイプのモデルでは、それぞれの桁に意味を持たせることもできます。たとえば、文化を宗教の教義と考えれば、各桁の数字によって「伝道の義務」、「他の教義に対する排他性」といった特徴の強さを表すことが出来ます。それを利用して、エージェントの行動を、持っている教義によって変化させることも可能です。今回は、1桁目を「他の教義に対する排他性」としてモデルに組み込んでみましょう。寛容さは0から9で表しましょう。相手と交流をしたときに、9であればいまままで通り100%の確率で相手の教義のいずれかの要素を受け入れますが、0であれば10%しか受け入れません。

***かなり上級(下のヒントを見ないで作ったらすごい)**

InStr()、Len()を使って、重複を除いた文化の数を時系列グラフに出力してください。引数などはヘルプを参照して自分で調べてみてください。(それも練習のうちです)

のヒント:

あくまでも一つのやり方ですが、文化の桁に無意味な文字を一つ足すと楽です。文字列型の変数を作る エージェントを全てとってくる そのエージェントの文化がその変数に含まれていなければ、その変数の最後にエージェントの文化を追加する 123_332_153_347_968_465_...といった風に、重複のない長いリストができる その全体の長さを数えて割り算する それを表示する。という手があります。回答例もアップロードしますので、そちらも見てみてください。