

BB3 (状態数が3の Busy Beaver. Busy Beaver というのは空白のテープに同じ文字を有限個書くプログラム)

- これは "C言語による計算の理論(鹿島亮 著)" の4つの例題(suc, EQ_{ab}, BB3, Runner) の1つです.
- "C言語による計算の理論" の Turing machine は一度に「文字を1つ書くこと」「左右に1つ進むこと(または動かないこと)」の2つの事ができます.
- 「2進法」も「aやbの個数」も使います. 終了する時は数字列の左端に移動する必要はありません.
- 遷移図は 上記の本からお借りしました.
- TuringMachineの仕様は TuringMachine[rules,初期状態,step数] となります.
ruleは{q,s}→{q',s',dir} (q:現在の状態,s:ヘッドの下文字,q':次の状態,s':ヘッドが書き込む文字,dir:ヘッドの進む方向.右,左,留の3通りで,+1,-1,0で指定)
初期状態は{{q0,pos},tape} (q0:最初の状態,pos:最初のヘッドの位置,tape:最初のテープの状態) となります.
- 例えば{1,a}→{2,b,+1} は「状態1でheadの文字がa」なら「ヘッドの下に文字bを書いて,ヘッドは右に1つ進み,内部状態は状態2へ移る」という事です.
- 同じ例は遷移図の方では「(1の書いてある丸)→(ab右)→(2の書いてある丸)」と表されます.
- tapeの初期状態は変数[tape]に入っています. 初期状態は変えることができますが,両端に空白がそれぞれ2個以上「最後まで残る」様にして下さい.
- 最後のManipulateではテープをクリックしてdragすると,画面の大きさが変わります.

■ 補助program

```
ClearAll["Global`*"]
```

(*ヘッドの位置を[]で,状態は添字で表示する*)

```
qbracket[x_List] := (*状態+ヘッドの位置.プログラムの検証にGood*)
```

```
ReplacePart[x[[2]], x[[1]][[2]] → Subscript["[" <> ToString[x[[2]][[x[[1]][[2]]]] <> "]", x[[1], 1]]]
```

(*状態FがTuringMachine[]の出力=outに最初に現れる位置.*)

```
posF[x_] := FirstPosition[x[[All, 1, 1]], F][[1]] - 1
```

```
turing[out_] :=
```

```
Manipulate[
```

```
Block[{now, tape, control, pos, state, contents, boxes},
```

```
now = out[[step];
```

```
tape = now[[2];
```

```
control = now[[1];
```

```
pos = control[[2];
```

```
state = control[[1];
```

```
boxes = {Line[{{1, 0}, {1, 2}}], Table[Line[{{i, 0}, {i + 1, 0}, {i + 1, 2}, {i, 2}}], {i, 1, Length[tape]}};
```

```
contents = {Table[Text[Style[tape[[i]], Large], {i + .5, 1}], {i, 1, Length[tape]}],
```

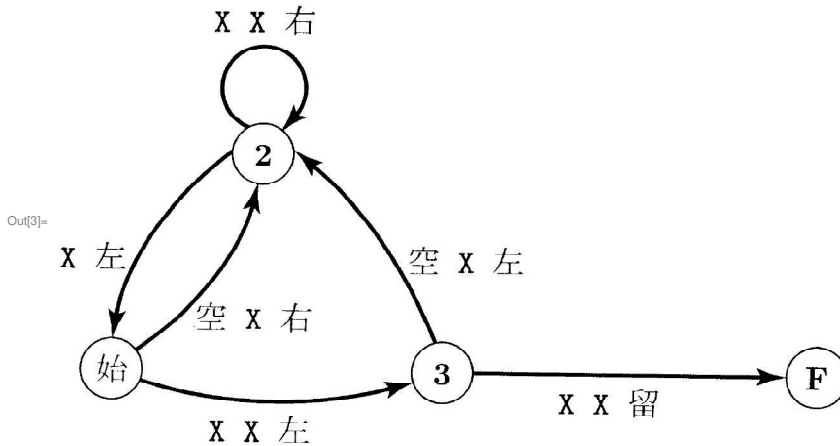
```
Green, Polygon[{{pos, -1}, {pos + 1, -1}, {pos + 1, -.5}, {pos + .5, 0}, {pos, -.5}],
```

```
Black, Text[Style[state, Medium], {pos + 0.5, -0.7}];
```

```
Graphics[{{boxes, contents}}, {step, 1, Length[out], 1}]
```

main program

In[3]:= Import["https://mixedmoss.com/mathematica/turing/jpg/BB3.jpg"]



(*漢字, カタカタでも大丈夫. しかしC,D,E は予約語なので, "C", "D", "E"としないといけない*)

```
rule = {{s, " " } -> {2, X, +1}, {s, X} -> {3, X, -1},
        {2, X} -> {2, X, +1}, {2, " " } -> {s, X, -1}, {3, " " } -> {2, X, -1}, {3, X} -> {F, X, 0}};
```

```
tape = {" ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " "};
```

```
bb3 = TuringMachine[rule, {{s, 5}, tape}, 13];
```

```
qbracket /@%
```

```
{{ , , , [ ]s, , , , }, { , , , X, [ ]2, , , , },
 { , , , [X]s, X, , , , }, { , , , [ ]3, X, X, , , , }, { , , [ ]2, X, X, X, , , , },
 { , [ ]s, X, X, X, X, , , , }, { , X, [X]2, X, X, X, , , , }, { , X, X, [X]2, X, X, , , , },
 { , X, X, X, [X]2, X, , , , }, { , X, X, X, X, [X]2, , , , }, { , X, X, X, X, X, [ ]2, , , , },
 { , X, X, X, X, [X]s, X, , , , }, { , X, X, X, [X]3, X, X, , , , }, { , X, X, X, [X]F, X, X, , , , }}
```

```
turing[bb3]
```

