

Minimum(左の数字と右の数字の最小値)

- ここに上げたのは私が2021年の春に"computability and logic"の本を読んだときに興味を持って作ったプログラムです。一部は練習問題でしたが、解答がないので特に書いてなければ自作です。
- "computability and logic" の Turing machine は一度に一つのことしかできません。Turing machine の中でも原始的なマシンです。例えば「1 を書いて右に動く」操作は、「1 を書く」 & 「右に動く」の2つの操作になります。よって状態数が大きくなります。また文字は "1" と "0(空白)" し使いません。
- 「2進法」ではなく「1の個数」で計算します。さらに数字列の左端に移動して終了するのでその為にも状態数が大きくなります。
- 状態の数が多いため遷移図は省略します。(^_^) プログラムの「rule」を御覧ください。
- 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:ヘッドの下の文字,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。
Mathematicaをお持ちの方はもちろんHelpを御覧ください。
- 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

```
In[103]:= ClearAll["Global`*"]
```

```
In[104]:= (*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[107]:= rule = {{1, " " } → {2, " ", 1}, {1, 1} → {1, " ", 0}, {2, " " } → {20, " ", 0}, {2, 1} → {3, 1, 0}, {3, " " } → {4, " ", 1},  
{3, 1} → {3, 1, 1}, {4, " " } → {4, " ", 1}, {4, 1} → {5, " ", 0}, {5, " " } → {6, " ", 1}, {5, 1} → {6, 1, 1},  
{6, " " } → {30, " ", 0}, {6, 1} → {7, 1, 0}, {7, " " } → {8, " ", 1}, {7, 1} → {7, 1, 1}, {8, " " } → {9, 1, 0},  
{8, 1} → {8, 1, 1}, {9, " " } → {10, " ", 0}, {9, 1} → {9, 1, -1}, {10, " " } → {10, " ", -1}, {10, 1} → {11, 1, 0},  
{11, " " } → {12, " ", 0}, {11, 1} → {11, 1, -1}, {12, " " } → {12, " ", -1}, {12, 1} → {13, 1, -1},  
{13, " " } → {1, " ", 1}, {13, 1} → {13, 1, -1}, {20, " " } → {20, " ", 1}, {20, 1} → {21, " ", 0},  
{21, " " } → {22, " ", 1}, {21, 1} → {21, " ", 0}, {22, " " } → {F, 1, 0}, {22, 1} → {21, 1, 0}, {30, " " } → {30, 1, 0},  
{30, 1} → {31, 1, -1}, {31, " " } → {31, " ", -1}, {31, 1} → {32, 1, 0}, {32, " " } → {33, " ", -1},  
{32, 1} → {32, " ", 0}, {33, " " } → {34, " ", 1}, {33, 1} → {32, 1, 0}, {34, " " } → {34, " ", 1}, {34, 1} → {F, 1, 0}};
```

```
In[108]:= tape = PadRight[PadLeft[{1, 1, 1, 1, 1, " ", 1, 1, 1, 1}, 20, " "], 30, " "]
```

```
pos = FirstPosition[tape, 1][[1]];
```

```
Out[108]= { , , , , , , , , , 1, 1, 1, 1, 1, , 1, 1, 1, , , , , , , , , }
```

```
In[110]:= minimum = TuringMachine[rule, {{1, pos}, tape}, 89];
```

```
qbracket /@ minimum  
(*posF[minimum]*)
```

```
Out[111]= {{ , , , , , , , , , [1]_1, 1, 1, 1, 1, , 1, 1, 1, , , , , , , , , },  
{ , , , , , , , , , [ ]_1, 1, 1, 1, 1, , 1, 1, 1, , , , , , , , , },
```


