

# Subst<sub>b</sub><sup>a</sup> (a,b,cからなる文字列で全てのbを消去し間を詰め,さらにaの個数と同じだけのbを左端に加える. 例えば「aabacbba→bbbbaaaca」(代入b=a=5),「aca→bbaca」(代入b=a=2))

- ここに上げたのは私が2022年の春に「C言語による計算の理論(鹿島亮 著)」を読んだときに興味を持って作ったプログラムです. 第9章の章末問題でしたが, 解答がないので自作です.
- 「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」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.
- 状態の数が少ないので遷移図も書きました. 遷移図とプログラムの rules が異なる場合はプログラムのほうが正解です(^\_^)
- 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

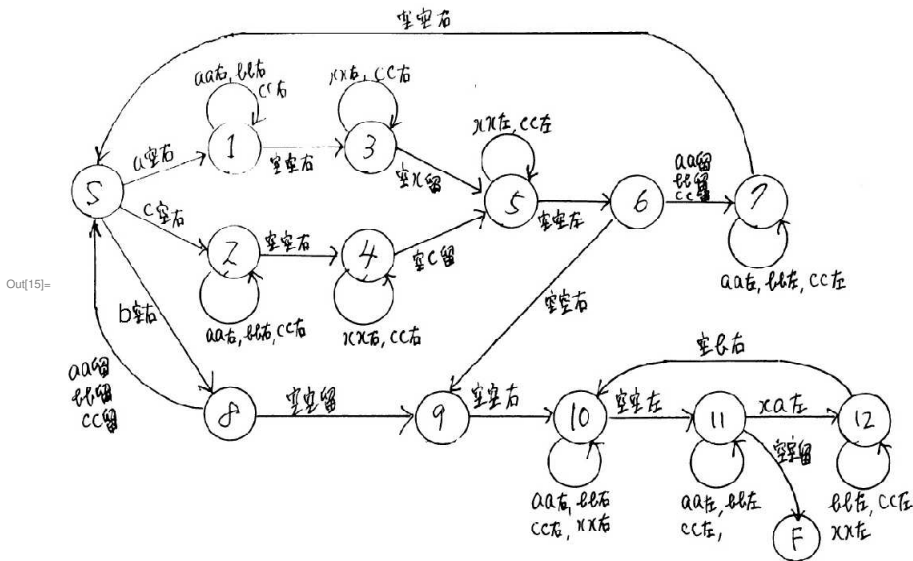
```
In[11]:= ClearAll["Global`*"]

In[12]:= (*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]]]
(*状態がFとなるstep数を求める*)
posF[x_] := FirstPosition[x[[All, 1, 1]], F][[1]] - 1

In[14]:= (*out には TuringMachine[]の出力が入る*)
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 = Graphics[Table[Line[{{i, 0}, {i + 1, 0}, {i + 1, 2}, {i, 2}, {i, 0}}], {i, 1, Length[tape]}]];
contents = Graphics[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}]];
Show[{boxes, contents}], {step, 1, Length[out], 1}]
```

## main program

```
In[15]= Import["https://mixedmoss.com/mathematica/substb2a.jpg"]
```



- [9]まではEraseAllbとほぼ同じ。ただし右側の文字列ではaの代わりにxを書く。[10]からはxをaに書き換え、文字列の左にbを加える

```
In[16]= rule = {{s, a} -> {1, " ", +1}, {s, c} -> {2, " ", +1}, {s, b} -> {8, " ", +1},
  {1, a} -> {1, a, +1}, {1, b} -> {1, b, +1}, {1, c} -> {1, c, +1}, {1, " "} -> {3, " ", +1},
  {2, a} -> {2, a, +1}, {2, b} -> {2, b, +1}, {2, c} -> {2, c, +1}, {2, " "} -> {4, " ", +1},
  {3, x} -> {3, x, +1}, {3, c} -> {3, c, +1}, {3, " "} -> {5, x, 0},
  {4, x} -> {4, x, +1}, {4, c} -> {4, c, +1}, {4, " "} -> {5, c, 0},
  {5, x} -> {5, x, -1}, {5, c} -> {5, c, -1}, {5, " "} -> {6, " ", -1},
  {6, a} -> {7, a, 0}, {6, b} -> {7, b, 0}, {6, c} -> {7, c, 0}, {6, " "} -> {9, " ", 1},
  {7, a} -> {7, a, -1}, {7, b} -> {7, b, -1}, {7, c} -> {7, c, -1}, {7, " "} -> {s, " ", +1},
  {8, a} -> {s, a, 0}, {8, b} -> {s, b, 0}, {8, c} -> {s, c, 0}, {8, " "} -> {9, " ", 0},
  {9, " "} -> {10, " ", +1},
  {10, a} -> {10, a, +1}, {10, b} -> {10, b, +1}, {10, c} -> {10, c, +1}, {10, x} -> {10, x, +1}, {10, " "} -> {11, " ", -1},
  {11, a} -> {11, a, -1}, {11, b} -> {11, b, -1}, {11, c} -> {11, c, -1}, {11, x} -> {12, a, -1}, {11, " "} -> {F, " ", 0},
  {12, b} -> {12, b, -1}, {12, c} -> {12, c, -1}, {12, x} -> {12, x, -1}, {12, " "} -> {10, b, +1}
};
```



In[24]:= `turing[substb2a]`

