

EraseAll_b (a,b,cからなる文字列に対し,bを全て消去し間を詰める.例: aabacbc→aaacc, ac→ac, b→" ")

- 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[1]:= ClearAll["Global`*"]

In[2]:= (*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[4]:= (*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}

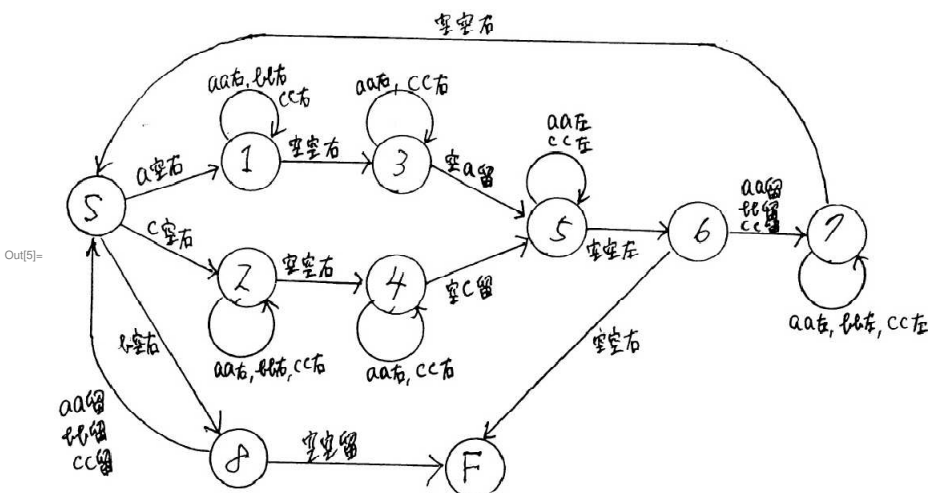
```

■ main program

```

In[5]:= Import["https://mixedmoss.com/mathematica/eraseAllb.jpg"]

```



- 次の文字がaやbである事を、状態([1]~[4])を使って記憶する点はErasebと同じだが、与えられた文字列を加工するのではなく、新しく文字列を作る。状態[8]は、bが右端にある時でも状態Fで終了させる為に追加した。無くても問題ない。(bが右端にある時に状態[1]で終了するだけ)

```

In[6]:= rules = {{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, a} -> {3, a, +1}, {3, c} -> {3, c, +1}, {3, " "} -> {5, a, 0},
  {4, a} -> {4, a, +1}, {4, c} -> {4, c, +1}, {4, " "} -> {5, c, 0},
  {5, a} -> {5, a, -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, " "} -> {F, " ", 1},
  {7, a} -> {7, a, -1}, {7, b} -> {7, b, -1}, {7, c} -> {7, c, -1}, {7, " "} -> {s, " ", +1},
  {8, " "} -> {F, " ", 0}, {8, a} -> {s, a, 0}, {8, b} -> {s, b, 0}, {8, c} -> {s, c, 0}};

In[7]:= tape = {" ", " ", a, a, b, a, c, b, b, a, " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " "};

eraseallb = TuringMachine[rules, {{s, 3}, tape}, 100];
qbracket /@ %

Out[9]:= {{ , , [a]s, a, b, a, c, b, b, a, , , , , , , },
  { , , [a]1, b, a, c, b, b, a, , , , , , , }, { , , , a, [b]1, a, c, b, b, a, , , , , , , },
  { , , , a, b, [a]1, c, b, b, a, , , , , , , }, { , , , a, b, a, [c]1, b, b, a, , , , , , , },
  { , , , a, b, a, c, [b]1, b, a, , , , , , , }, { , , , a, b, a, a, c, b, [b]1, a, , , , , , , },
  { , , , a, b, a, c, b, b, [a]1, , , , , , , }, { , , , a, b, a, a, c, b, b, a, [ ]1, , , , , , , },
  { , , , a, b, a, c, b, b, a, [ ]3, , , , , , , }, { , , , a, b, a, c, b, b, a, , [a]5, , , , , , , },
  { , , , a, b, a, c, b, b, a, [ ]5, a, , , , , , , }, { , , , a, b, a, c, b, b, [a]6, , a, , , , , , , },
  { , , , a, b, a, c, b, b, [a]7, , a, , , , , , , }, { , , , a, b, a, c, b, [b]7, a, , a, , , , , , , },
  { , , , a, b, a, c, [b]7, b, a, , a, , , , , , , }, { , , , a, b, a, [c]7, b, b, a, , a, , , , , , , },
  { , , , a, b, [a]7, c, b, b, a, , a, , , , , , , }, { , , , a, [b]7, a, c, b, b, a, , a, , , , , , , },
  { , , , [a]7, b, a, c, b, b, a, , a, , , , , , , }, { , , , [ ]7, a, b, a, c, b, b, a, , a, , , , , , , },
  { , , , [a]5, b, a, c, b, b, a, , a, , , , , , , }, { , , , , [b]1, a, c, b, b, a, , a, , , , , , , },
  { , , , , b, [a]1, c, b, b, a, , a, , , , , , , }, { , , , , b, a, [c]1, b, b, a, , a, , , , , , , },
  { , , , , b, a, c, [b]1, b, a, , a, , , , , , , }, { , , , , b, a, c, b, [b]1, a, , a, , , , , , , },
  { , , , , b, a, c, b, b, [a]1, , a, , , , , , , }, { , , , , b, a, c, b, b, a, [ ]1, a, , , , , , , },
  { , , , , b, a, c, b, b, a, , [a]3, , , , , , , }, { , , , , b, a, c, b, b, a, , a, [ ]3, , , , , , , },
  { , , , , b, a, c, b, b, a, , a, [a]5, , , , , , , }, { , , , , b, a, c, b, b, a, , [a]5, a, , , , , , , },
  { , , , , b, a, c, b, b, a, [ ]5, a, a, , , , , , }, { , , , , b, a, c, b, b, [a]6, , a, a, , , , , , },
  { , , , , b, a, c, b, b, [a]7, , a, a, , , , , , , }, { , , , , b, a, c, b, [b]7, a, , a, a, , , , , , , },
  { , , , , b, a, c, [b]7, b, a, , a, a, , , , , , , }, { , , , , b, a, [c]7, b, b, a, , a, a, , , , , , , },
  { , , , , b, [a]7, c, b, b, a, , a, a, , , , , , , }, { , , , , [b]7, a, c, b, b, a, , a, a, , , , , , , },
  { , , , , [ ]7, b, a, c, b, b, a, , a, a, , , , , , , }, { , , , , [b]5, a, c, b, b, a, , a, a, , , , , , , },
  { , , , , , [a]8, c, b, b, a, , a, a, , , , , , , }, { , , , , , [a]5, c, b, b, a, , a, a, , , , , , , },
  { , , , , , [c]1, b, b, a, , a, a, , , , , , , }, { , , , , , c, [b]1, b, a, , a, a, , , , , , , },
  { , , , , , c, b, [b]1, a, , a, a, , , , , , , }, { , , , , , c, b, b, [a]1, , a, a, , , , , , , },
  { , , , , , c, b, b, a, [ ]1, a, a, , , , , , , }, { , , , , , c, b, b, a, , [a]3, a, , , , , , , },
  { , , , , , c, b, b, a, , a, [a]3, , , , , , , }, { , , , , , c, b, b, a, , a, a, [ ]3, , , , , , , },
  { , , , , , c, b, b, a, , a, a, [a]5, , , , , , , }, { , , , , , c, b, b, a, , a, a, [a]5, a, , , , , , , },
  { , , , , , c, b, b, a, , [a]5, a, a, , , , , , , }, { , , , , , c, b, b, a, [ ]5, a, a, a, , , , , , , },
  { , , , , , c, b, b, [a]6, , a, a, a, , , , , , , }, { , , , , , c, b, b, [a]7, , a, a, a, , , , , , , },
  { , , , , , c, b, [b]7, a, , a, a, a, , , , , , , }, { , , , , , c, [b]7, b, a, , a, a, a, , , , , , , },
  { , , , , , [c]7, b, b, a, , a, a, a, , , , , , , }, { , , , , , [ ]7, c, b, b, a, , a, a, a, , , , , , , },
  { , , , , , [c]5, b, b, a, , a, a, a, , , , , , , }, { , , , , , , [b]2, b, a, , a, a, a, , , , , , , },
  { , , , , , b, [b]2, a, , a, a, a, c, , , , , , , }, { , , , , , b, b, [a]2, , a, a, a, , , , , , , },
  { , , , , , b, b, a, [ ]2, a, a, a, , , , , , , }, { , , , , , b, b, a, , [a]4, a, a, , , , , , , },
  { , , , , , b, b, a, , a, [a]4, a, , , , , , , }, { , , , , , b, b, a, , a, a, [a]4, , , , , , , },
  { , , , , , b, b, a, , a, a, a, [ ]4, , , , , , , }, { , , , , , b, b, a, , a, a, a, [c]5, , , , , , , },
  { , , , , , b, b, a, , a, a, [a]5, c, , , , , , , }, { , , , , , b, b, a, , a, a, [a]5, a, c, , , , , , , },
  { , , , , , b, b, a, , [a]5, a, a, c, , , , , , , }, { , , , , , b, b, a, [ ]5, a, a, a, c, , , , , , , },
  { , , , , , b, b, [a]6, , a, a, a, c, , , , , , , }, { , , , , , b, b, [a]7, , a, a, a, c, , , , , , , },
  { , , , , , b, [b]7, a, , a, a, a, c, , , , , , , }, { , , , , , [b]7, b, a, , a, a, a, c, , , , , , , },
  { , , , , , [ ]7, b, b, a, , a, a, a, c, , , , , , , }, { , , , , , [b]5, b, a, , a, a, a, c, , , , , , , },
  { , , , , , [b]8, a, , a, a, a, c, , , , , , , }, { , , , , , [b]5, a, , a, a, a, c, , , , , , , },
  { , , , , , [a]8, , a, a, a, c, , , , , , , }, { , , , , , [a]5, , a, a, a, c, , , , , , , },
  { , , , , , [ ]1, a, a, a, c, , , , , , , }, { , , , , , [a]3, a, a, c, , , , , , , },
  { , , , , , a, [a]3, a, c, , , , , , , }, { , , , , , a, a, [a]3, c, , , , , , , },
  { , , , , , a, a, a, [c]3, , , , , , , }, { , , , , , a, a, a, c, [ ]3, , , , , , , },
  { , , , , , a, a, a, c, [a]5, , , , , , , }, { , , , , , a, a, a, [c]5, a, , , , , , , },
  { , , , , , a, a, [a]5, c, a, , , , , , , }, { , , , , , a, [a]5, a, c, a, , , , , , , },
  { , , , , , [a]5, a, a, c, a, , , , , , , }, { , , , , , [ ]5, a, a, a, c, a, , , , , , , },
  { , , , , , [ ]6, a, a, a, c, a, , , , , , , }, { , , , , , [ ]F, a, a, a, c, a, , , , , , , }


```

In[10]:= `turing[eraseallb]`

step

Out[10]=

		a	a	b	a	c	b	b	a										
--	--	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--

 s

