John Spicer
I pledge my honor that I have abided by the Stevens Honor System
11/11/15

Name: Switch

```
init: qInit
accept: qAccept

qInit,0,_
qCopy#,0,_,-,-

qInit,1,_
qCopy#,1,_,-,-

qCopy#,0,_
qCopy#,0,_,>,-

qCopy#,1,_
qCopy#,1,_,>,-

qCopy#,#,_
qTest,x,_,>,-

qTest,0,_
qDo,0,_,<,-

qTest,1,_
qDo,1,_,<,-

qDo,0,_
qDo,0,_,<,-

qDo,1,_
qDo,1,_,<,-

qDo,x,_
qCopy0,x,_,>,-

qCopy0,0,_
qCopy0,0,0,>,>

qCopy0,1,_
qCopy0,1,1,>,>

qCopy0,_,_
qBegin,_,#,<,>
```

```
q,1,_
qBegin,1,_,<,-

qBegin,0,_
qBegin,0,_,<,-

qBegin,x,_
qBegin,x,_,<,-

qBegin,_,_
qCopy1,_,_,>,-

qCopy1,1,_
qCopy1,1,1,>,>

qCopy1,0,_
qCopy1,0,0,>,>

qCopy1,x,_
qAccept,_,_,-,-


name: antipalin

init: qCopy
accept: qAccept

qCopy,0,_
qCopy,0,0,>,>

qCopy,1,_
qCopy,1,1,>,>

qCopy,_,_
qBack,_,_,<,-

qBack,0,_
qBack,0,_,<,-

qBack,1,_
qBack,1,_,<,-

qBack,_,_
qTest,_,_,>,<

qTest,1,1
qTest,1,1,>,<

qTest,0,0
```

```
qTest,0,0,>,<

qTest,1,0
qAccept,1,0,-,-

qTest,0,1
qAccept,0,1,-,-


name: equals

init: q0
accept: qAccept

qzer,0
qLook0,x,<

qzer,1
qzer,1,>

qzer,x
qzer,x,>

qzer,_
q2,_,<

qLook0,0
qLook0,0,<

qLook0,1
qLook0,1,<

qLook0,x
qLook0,x,<

qLook0,_
q1,_,>

q1,x
q1,x,>

q1,0
q1, 0,>

q1,_
q2,_,<

q1,1
qLook1,x,<
```

qLook1,0
qLook1,0,<

qLook1,1
qLook1,1,<

qLook1,x
qLook1,x,<

qLook1,_
q0,_,>

q2,0
q2,0,<

q2,x
q2,x,<

q2,1
q2,1,<

q2,_
q3,_,>

q3,x
q3,x,>

q3,_
qAccept,_,-

John Spicer

32. b. $1\#1$ q_1

→ reads 1, writes x, move right

$x\#1$ q_3

→ reads #, moves right

$x\#1$ q_5

→ reads 1, moves ~~right~~ left, writes x

$x\#x$ q_6

→ reads #, moves left

$x\#x$ q_7

→ reads x, moves right

$x\#x$ q_1

→ reads #, moves right

$x\#x$ q_8

→ reads x, moves right

$x\#x \sqcup$ q_8

reads ⊔, accept state q accepted

c. $1\#\#1$ q_1

→ reads 1, writes x, move R

$x\#\#1$, q_3

→ reads #, move R

$x\#\#1$, q_5

reads #, reject, q reject

d. $\underline{1}0\#11$ q_1

→ reads 1, write x, move R

$x\underline{0}\#11$ q_3

→ reads 0, move R

$x0\underline{\#}11$ q_3

→ reads #, move R

$x0\#\underline{1}1$ q_3

→ reads 1, writes x, moves L

$x0\underline{\#}x1$ q_6

→ reads #, move L

$x\underline{0}\#x1$ q_7

→ reads 0, move L

$\underline{x}0\#x1$ q_7

→ reads x, move R

$x\underline{0}\#x1$ q_1

→ read 0, write x, move R

$xx\underline{\#}x1$ q_2

→ read #, move R

$xx\#\underline{x}1$ q_4

→ read x, move R

$xx\#x\underline{1}$ q_4

→ read 1, reject q_{reject}

e. 1 0 # 1 0 a_1

→ read 1, write x, move R

x 0 # 1 0 a_3

→ read 0 move R

x 0 # 1 0 a_3

→ read #, move right

x 0 # 1 0 a_5

→ read 1, write x, move L

x 0 # x 0 a_6

→ read #, move L

x 0 # x 0 a_7

→ read 0, move L

x 0 # x 0 a_7

→ read x, move R

x 0 # x 0 a_1

→ read 0, write x, move R

x x # x 0 a_2

→ read #, move R

x x # x 0 a_4

→ read x, move R

x x # x 0 a_4

→ read 0, move ~~L~~ right x

x x # x x a_6

→ read x, move L

x x # x x a_6

→ read #, move L

x x # x x a_7

→ read x, move R

x x # x x a_1

→ read #, move R

x x # x x a_8 ↓

→ read x, move R
xx#xx̲ q_8
→ read x, move R
xx#xx___
→ read ⊔, accept, q accept

3.8. b. {w|w contains twice as many 0's as 1's}
Stage 1:
mark first unmarked 0, if none goto 5
Stage 2:
mark the next unmarked 0, if no more 0, reject, else goto beginning
Stage 3:
mark first unmarked 0, if none, reject
Stage 4:
move to front, repeat Stage 1
Stage 5:
move head to front, see if any unmarked 0's or 1's, if none, accept, else reject

c. {w|w does not contain twice as many zeros as ones}
Stage 1: mark first unmarked 0, if none goto 5
Stage 2: mark next unmarked 0, if none reject, else move to front
Stage 3: mark first unmarked 1, if none reject
Stage 4: move head to front, goto Stage 1
Stage 5: move head to front, ~~goto~~ scan for unmarked 1's, if none reject, else accept

3.12 Let M be a Turing Machine
and M_L be the TM w/ left reset M_L
M: ~~symbols~~ simulates

M_L makes R transition when M makes the same

- when M make a R transition M_L follows it
- when M makes L transition with a or b in M, M_L replaces it with A/B
  so $\Sigma_{M_L} = \{A_n\} \cup \{A, B\}$ & resets left.
  shifts all to the R for everything except $\{A, B\}$

Process repeated until all is RsWtted
M_L does a reset. All right transitions are checked as M does.

3.15b. L_1 & L_2 2 decidable langs.
M_1 & M_2 are TMs that decide them.
Show: $L(M') = L_1 \circ L_2$

M' = " on input w

1. $w \to w_1, w_2 \Rightarrow w = w_1 w_2$
2. M_1 on w_1, if M_1 rejects then reject
3. Else run M_2 on w_2 if M_2 reject then reject
4. Else accept

Decidable languages are closed under contatenation

c. L is decidable by M
M' decides L^*
M', on input w
1. split w in n parts
2. Run M on w_i for $i = 1, 2, \ldots n$
If M accepts each, accept
3. all have been tried w/o success, reject

$L(M') = L^*$
decidable langs closed under *

d. TM M decides lang A
TM M' decides compliment of A.

On input w
1. simulate M on w (a) if M accepts, w, reject (b) if M rejects, accept

Since M halts, M' halts, M' decides compliment of A.

Decideable langs closed under complementation

L_1 + L_2 are decidable by M_1 + M_2
1. show $L(M') = L_1 \cap L_2$
2. $M' =$ "on input w:
1. run M_1 on w, if M_1 rejects, reject
2. Else run M_2 on w_1, if M_2 rejects, reject
3. Else accept"

$L(M') = L_1 \cap L_2$