Solution to Problem 13

Problem. Similarly to a Turing machine that we had in class, that copies a number in unary code, design a Turing machine that copies words in a 2-letter alphabet {a,b}. Test it on the example of a word bab. The result should be bab_bab, with a blank space in between.

Hint: instead of marking 1s, mark both a's and b's; instead of the state carry1in1st, we can now have two different states: carryAin1st and carryBin1st.

Solution. First, we start moving:

- start, – $\to$ R, in1st

If we see blank, this means that we had nothing to copy – the original word was an empty string. So, we go back and halt.

- in1st, – $\to$ L, halt

If we see a or b, we mark it and go to the state carryAin1st or carryBin1st:

- in1st, a $\to$ $\widehat{\text{a}}$, R, carryAin1st
- in1st, b $\to$ $\widehat{\text{b}}$, R, carryBin1st

We move step by step inside the 1st (original) number:

- carryAin1st, a $\to$ R
- carryAin1st, b $\to$ R
- carryBin1st, a $\to$ R
- carryBin1st, b $\to$ R

Once we reach a blank space, we know that after moving one step to the right we will be in the 2nd number:

- carryAin1st, – $\to$ R, carryAin2nd
- carryBin1st, – $\to$ R, carryBin2nd

As long as we see a's and b's, we continue going through the 2nd number:

- carryAin2nd, a $\to$ R

- carryAin2nd, b $\rightarrow$ R
- carryBin2nd, a $\rightarrow$ R
- carryBin2nd, b $\rightarrow$ R

Once we see the first blank space, we drop the carried bit there and start going back:

- carryAin2nd, $-$ $\rightarrow$ a, L, backIn2nd
- carryBin2nd, $-$ $\rightarrow$ b, L, backIn2nd

We go left through the second number:

- backIn2nd, a $\rightarrow$ L
- backIn2nd, b $\rightarrow$ L

Once we read the blank space separating the second number from the first one, we need to check if we are done:

- backIn2nd, $-$ $\rightarrow$ L, checkIfDone

If the symbol that we see in the 1st number is an unmarked symbol, this means that we are not done, so we need to go back and start finding the first unmarked symbols:

- checkIfDone, a $\rightarrow$ L, backIn1st
- checkIfDone, b $\rightarrow$ L, backIn1st

As we go left, if we see an unmarked symbol, we continue going left:

- backIn1st, a $\rightarrow$ L
- backIn1st, b $\rightarrow$ L

Once we meet a marked symbol, this means that next one to the right if the first unmarked on. So we go to the state in1st to repeat the whole procedure:

- backIn1st, $\widehat{\mathrm{a}}$ $\rightarrow$ R, in1st
- backIn1st, $\widehat{\mathrm{b}}$ $\rightarrow$ R, in1st

If the first symbol we see after getting into the 1st number is marked, this means that there are no more unmarked symbols in the 1st number. So, we unmark the marked symbol that we see and go to the unmark state:

- checkIfDone, $\widehat{\mathrm{a}}$ $\rightarrow$ a, L, unmark
- checkIfDone, $\widehat{\mathrm{b}}$ $\rightarrow$ b, L, unmark

Then, we go left step by step and unmark all the symbols of the first number one by one:

- unmark, $\widehat{a} \to a$, L
- unmark, $\widehat{b} \to b$, L

Once we reach the very first (blank) cell of the Turing machine, this means that we are done. So we halt:

- unmark, $- \to$ halt

Let us trace this Turing machine on the example of the word 1011.

$\underline{-}$	b	a	b	–	–	–	–	–	–	–	...	start
–	$\underline{b}$	a	b	–	–	–	–	–	–	–	...	in1st
–	$\widehat{b}$	$\underline{a}$	b	–	–	–	–	–	–	–	...	carryBin1st
–	$\widehat{b}$	a	$\underline{b}$	–	–	–	–	–	–	–	...	carryBin1st
–	$\widehat{b}$	a	b	$\underline{-}$	–	–	–	–	–	–	...	carryBin1st
–	$\widehat{b}$	a	b	–	$\underline{-}$	–	–	–	–	–	...	carryBin2nd
–	$\widehat{b}$	a	b	–	b	–	–	–	–	–	...	backIn2nd
–	$\widehat{b}$	a	$\underline{b}$	–	b	–	–	–	–	–	...	checkIfDone
–	$\widehat{b}$	$\underline{a}$	b	–	b	–	–	–	–	–	...	backIn1st
–	$\underline{\widehat{b}}$	a	b	–	b	–	–	–	–	–	...	backIn1st
–	$\widehat{b}$	$\underline{a}$	b	–	b	–	–	–	–	–	...	in1st
–	$\widehat{b}$	$\widehat{a}$	$\underline{b}$	–	b	–	–	–	–	–	...	carryAin1st
–	$\widehat{b}$	$\widehat{a}$	b	$\underline{-}$	b	–	–	–	–	–	...	carryAin1st
–	$\widehat{b}$	$\widehat{a}$	b	–	$\underline{b}$	–	–	–	–	–	...	carryAin2nd
–	$\widehat{b}$	$\widehat{a}$	b	–	b	$\underline{-}$	–	–	–	–	...	carryAin2nd
–	$\widehat{b}$	$\widehat{a}$	b	–	$\underline{b}$	a	–	–	–	–	...	backIn2nd
–	$\widehat{b}$	$\widehat{a}$	b	$\underline{-}$	b	a	–	–	–	–	...	backIn2nd
–	$\widehat{b}$	$\widehat{a}$	$\underline{b}$	–	b	a	–	–	–	–	...	checkIfDone
–	$\widehat{b}$	$\underline{\widehat{a}}$	b	–	b	a	–	–	–	–	...	backIn1st
–	$\widehat{b}$	$\widehat{a}$	$\underline{b}$	–	b	a	–	–	–	–	...	in1st
–	$\widehat{b}$	$\widehat{a}$	$\widehat{b}$	–	b	a	–	–	–	–	...	carryBin1st

$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$\underline{-}$	b	a	$-$	$-$	$-$	$-$	...	carryBin1st
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$-$	$\underline{\text{b}}$	a	$-$	$-$	$-$	$-$	...	carryBin2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$-$	b	$\underline{\text{a}}$	$-$	$-$	$-$	$-$	...	carryBin2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$-$	b	a	$\underline{-}$	$-$	$-$	$-$	...	carryBin2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$-$	b	$\underline{\text{a}}$	b	$-$	$-$	$-$	...	backIn2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$-$	$\underline{\text{b}}$	a	b	$-$	$-$	$-$	...	backIn2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\widehat{\text{b}}$	$\underline{-}$	b	a	b	$-$	$-$	$-$	...	backIn2nd
$-$	$\widehat{\text{b}}$	$\widehat{\text{a}}$	$\underline{\widehat{\text{b}}}$	$-$	b	a	b	$-$	$-$	$-$	...	checkIfDone
$-$	$\widehat{\text{b}}$	$\underline{\widehat{\text{a}}}$	b	$-$	b	a	b	$-$	$-$	$-$	...	unmark
$-$	$\underline{\widehat{\text{b}}}$	a	b	$-$	b	a	b	$-$	$-$	$-$	...	unmark
$\underline{-}$	b	a	b	$-$	b	a	b	$-$	$-$	$-$	...	unmark
$\underline{-}$	b	a	b	$-$	b	a	b	$-$	$-$	$-$	...	halt