Lecture #6: Introduction to Turing Machines

What To Do Before the Lecture

1. Watch the videos for Lecture #6 —noting that they will probably be understandable if you play them at double speed. If you do not have time for this then look at the "Key Concepts" document that is found, immediately after the videos for this lecture, on the course web site, instead.
2. ***Print*** and read through the rest of this document and — if you have time — try to solve the problems! These should help you to check that you understand how Turing machines process strings, and that you understand how to prove things about them.
3. The supplemental materials include additional information about several things that are introduced in the lecture videos, but not explained in detail.

Problems To Be Solved

Continuing the First Example

Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and consider, once again, a Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

such that

$$Q = \{q_0, q_{\mathtt{a},1}, q_{\mathtt{a},2}, q_{\mathtt{b},1}, q_{\mathtt{b},2}, q_{\text{accept}}, q_{\text{reject}}\},$$

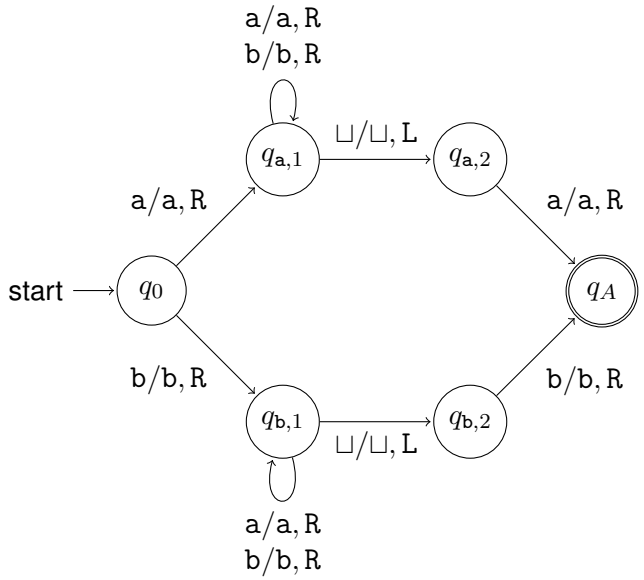
$\Gamma = \{\mathtt{a}, \mathtt{b}, \sqcup\}$, and the transition function, δ, is given by the following transition table:

	$\mathtt{a}$	$\mathtt{b}$	$\sqcup$
q_0	$(q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})$	$(q_{\mathtt{b},1}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$
$q_{\mathtt{a},1}$	$(q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})$	$(q_{\mathtt{a},1}, \mathtt{b}, \mathtt{R})$	$(q_{\mathtt{a},2}, \sqcup, \mathtt{L})$
$q_{\mathtt{a},2}$	$(q_{\text{accept}}, \mathtt{a}, \mathtt{R})$	$(q_{\text{reject}}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$
$q_{\mathtt{b},1}$	$q_{\mathtt{b},1}, \mathtt{a}, \mathtt{R})$	$q_{\mathtt{b},1}, \mathtt{b}, \mathtt{R})$	$(q_{\mathtt{b},2}, \sqcup, \mathtt{L})$
$q_{\mathtt{b},2}$	$q_{\text{reject}}, \mathtt{a}, \mathtt{R})$	$(q_{\text{accept}}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$

This Turing machine can also be described using the following picture. To keep the picture simple the accepting state is shown as "q_A" instead of q_{accept}, and transitions to the rejecting state, and this state, are left out — but, as the transition table indicates,

$$\delta(q, \sigma) = (q_{\text{reject}}, \sigma, \mathtt{R})$$

for $q = q_0$ and $\sigma = \sqcup$, for $q = \mathtt{a}, 2$ and either $\sigma = \mathtt{b}$ or $\sigma = \sqcup$, and for $q_{\mathtt{b},2}$ and either $\sigma = \mathtt{a}$ or $\sigma = \sqcup$.



1. The goal of the first part of this lecture presentation is to sketch a proof that this Turing machine ***decides*** the language

$$\begin{aligned} L &= \{\omega \in \{\mathtt{a}, \mathtt{b}\}^\star \mid |\omega| \geq 1 \text{ and } \omega \text{ begins, and ends, with the same letter}\} \\ &= \{\mathtt{a}, \mathtt{b}\} \cup \{\mathtt{a}\mu\mathtt{a} \mid \mu \in \{\mathtt{a}, \mathtt{b}\}^\star\} \cup \{\mathtt{b}\mu\mathtt{b} \mid \mu \in \{\mathtt{a}, \mathtt{b}\}^\star\}. \end{aligned}$$

   (a) Give a ***trace of execution*** to show that M ***rejects*** the empty string, λ:

   Trace of Execution:

(b) Give a ***trace of execution*** to show that M ***accepts*** aa.

Trace of Execution:

(c) Give a ***trace of execution*** to show that M ***accepts*** bb.

Trace of Execution:

Consider a string

$$\omega = \mathtt{a}\,\alpha_1\alpha_2\ldots\alpha_n \in \Sigma^\star$$

where $n \geq 1$ and $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Sigma$ (so that $|\omega| = n+1$).

Consider the following:

Lemma 1. *For every integer i, if $0 \leq i \leq n$ then*

$$q_0\omega \vdash^\star \mathtt{a}\,\alpha_1\alpha_2\ldots\alpha_i\,q_{\mathtt{a},1}\,\alpha_{i+1}\alpha_{i+1}\ldots\alpha_n.$$

Note: This asserts that

$$q_0\omega \vdash^\star \mathtt{a}\,q_{\mathtt{a},1}\alpha_1\alpha_2\ldots\alpha_n$$

(because this is what is claimed for the case that $i = 0$). It also asserts that

$$q_0\omega \vdash^\star \mathtt{a}\,\alpha_1\alpha_2\ldots,\alpha_n\,q_{\mathtt{a},1}$$

(because this is what is claimed for the case that $i = n$).

(d) Prove Lemma 1.

Proof of Lemma:

Lemma 2. *Suppose that*

$$\omega = \mathtt{a}\,\mu\,\mathtt{a}$$

for a string $\mu \in \Sigma^\star$*. Then* M **accepts** ω*.*

(e) Prove Lemma 2.

Proof of Lemma:

Lemma 3. *Suppose that*

$$\omega = \mathtt{a}\,\mu\,\mathtt{b}$$

for a string $\mu \in \Sigma^\star$*. Then* M **rejects** ω*.*

(f) Prove Lemma 3.

Proof:

Now consider a string

$$\omega = \mathsf{b}\,\alpha_1\alpha_2\ldots\alpha_n \in \Sigma^\star$$

where $n \geq 1$ and $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Sigma$ (so that $|\omega| = n + 1$).

(g) State, and prove, another lemma, "Lemma 4", that resembles Lemma 1 — but that concerns the beginning of M's execution on a string that begins with "b" (like the above one) instead of a string that begins with "a".

Lemma 4:

Proof of Lemma 4:

Lemma 5. *Suppose that*

$$\omega = \mathtt{b}\,\mu\,\mathtt{b}$$

for a string $\mu \in \Sigma^\star$*. Then* M **accepts** ω*.*

(h) Use Lemma 4 to prove Lemma 5.

Proof

Lemma 6. *Suppose that*

$$\omega = \mathtt{b}\,\mu\,\mathtt{a}$$

for a string $\mu \in \Sigma^\star$*. Then* M **rejects** ω*.*

(i) Use Lemma 4 to prove Lemma 6.

Proof:

Theorem 7. *M decides the language L.*

(j) Use the above results to prove Theorem 7.

Proof:

```
boolean isInL ( ω : Σ* ) {
1.  if (ω == λ) {
2.    MUST BE COMPLETED
3.  } else if (ω begins with "b") {
4.    MUST BE COMPLETED
5.  } else if (|ω| == 1) {
6.    MUST BE COMPLETED
7.  } else if (ω ends with "a") {
8.    MUST BE COMPLETED
    } else {
9.    Let μ ∈ Σ* such that ω = a · μ · b.
      MUST BE COMPLETED
    }
```

Figure 1: Incomplete Algorithm to Decide the Language $\widehat{L}$

Introduction to Turing Machine Design

It is recommended that, when designing a Turing machine for a given language by ***starting at a high level*** — picking an algorithm that can, eventually, by implemented using a Turing machine. It this not obvious you should continue by proving the correctness of the algorithm you will use.

2. Suppose that $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, and let

$$\widehat{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}\}$$

— where $\mathbb{N}$ represents the set of *non-negative* integers, so that $0 \in \mathbb{N}$ and $\lambda \in \widehat{L}$.

(a) Consider the — incomplete — algorithm shown in Figure 1. Give the pseudocode at lines 2, 4, 6, 8 and 9 that is needed to produce an algorithm that halts when executed on any input string $\omega \in \Sigma^\star$, returning `true` if $\omega \in \widehat{L}$ and returning `false` if $\omega \notin \widehat{L}$.

Solution:

(b) ***Prove*** that your algorithm correctly decides membership in $\widehat{L}$. That is, prove that an execution of the algorithm on any string $\omega \in \Sigma^\star$ always halts. Furthermore, this execution of the algorithm ends with `true` returned if $\omega \in \widehat{L}$, and it ends with `false` returned if $\omega \notin L$.

Proof of Correctness:

It turns out that completing the design of a Turing machine, that implements this algorithm, is simplified if we try to develop a Turing machine that satisfies some additional properties, as noted below:

i. Let q_0 be the start state. The Turing machine halts, after a finite number of steps, when the execution continues from configuration

$$\sqcup^h \, q_0 \, \omega \tag{1}$$

— for every non-negative integer h and for every string $\omega \in \Sigma^\star$.

ii. If $\omega \in \widehat{L}$ and the Turing machine continues an execution from the configuration shown at line (1), for any non-negative integer h, then the Turing machine reaches an ***accepting*** configuration after finitely many more steps.

iii. If $\omega \notin \widehat{L}$ and the Turing machine continues an execution from the configuration shown at line (1), for any non-negative integer h, then the Turing machine reaches a ***rejecting*** configuration after finitely many more steps.

(b) Prove that if a Turing machine M (with input alphabet Σ) satisfies properties i., ii., and iii., above, then M decides the language $\widehat{L}$.

Note: This should be easy!

Proof That These Properties are Sufficient:

(c) Use the above to give an ***implementation-level*** description of a Turing machine M, with input alphabet Σ, — and describe how this is based on the high-level algorithm that you completed at the beginning of this question. Your Turing machine should satisfy properties i., ii., and ii., above.

Note: Rather than writing some new symbol on top of symbols that you wish to eliminate, you should now be able to replace then with "$\sqcup$" — while completing an implementation-level description of a Turing machine that satisfies the properties that are given here.

An Implementation-Level Description — and Relationship to High-Level Algorithm:

(d) Use this to give a ***formal description*** of a Turing machine M that decides the language $\widehat{L}$ — and describe how it is related to (and based on) the implementation-level description that you gave before this.. Your Turing machine should satisfy properties i., ii., and iii., above.

Formal Description of M:

Relationship to Implementation-Level Description:

(e) Sketch a proof that M decides $\widehat{L}$ — by giving a sequence of lemmas that (somewhat) the resemble the ones in Question #1, above, and that can be used to prove correctness of M (just as the lemmas in Question #1 were used to give a reasonably simple proof of Theorem 7).

Ideally, these lemmas should not be hard to discover, because of the design process used here.

Sketch of Proof of Correctness: