**Air Force Institute of Technology**

**Graduate School of Engineering and Management**

**Department of Electrical and Computer Engineering**

**CSCE 532 Automata and Formal Languages**

**Winter 2019**

# Day 10 - The Church-Turing Thesis

§3.1 Turing Machines

### Definition

A **Turing machine** is a 7-tuple , where are all finite sets and

1. is the set of states,
2. is the input alphabet not containing the blank symbol ␣,
3. is the tape alphabet, where ␣ and ,
4. is the transition function,
5. is the start state,
6. is the accept state, and
7. is the reject state, where .

### Terminology and Notation

A **configuration** of a TM specifies its current state, tape contents, and head location. Specifically, for state and strings , the configuration specifies that the machine is in state , that the tape contains , and that the head is positioned over the first symbol of .

The **start configuration** of a TM with input is .

Any configuration involving the state () is an **accepting (rejecting) configuration**.

A **halting configuration** is one that is accepting or rejecting.

A configuration **yields** configuration if the TM can legally go from to in one step.

A TM **accepts** input if a sequence of configurations exists, where

1. is the start configuration of on input ,
2. Each yields , and
3. is an accepting configuration.

The collection of strings that accepts is the **language of** , or the **language recognized by** , denoted .

In general TMs can accept, reject, or loop. A TM that halts on all inputs is called a **decider**. A decider that recognizes some language is also said to **decide** it.

### Definition

Call a language **Turing-recognizable** (or **recursively enumerable**) if some TM recognizes it.

### Definition

Call a language **Turing-decidable** (or **decidable**, or **recursive**) if some TM decides it.

### Example (Sipser exercise 3.1C)

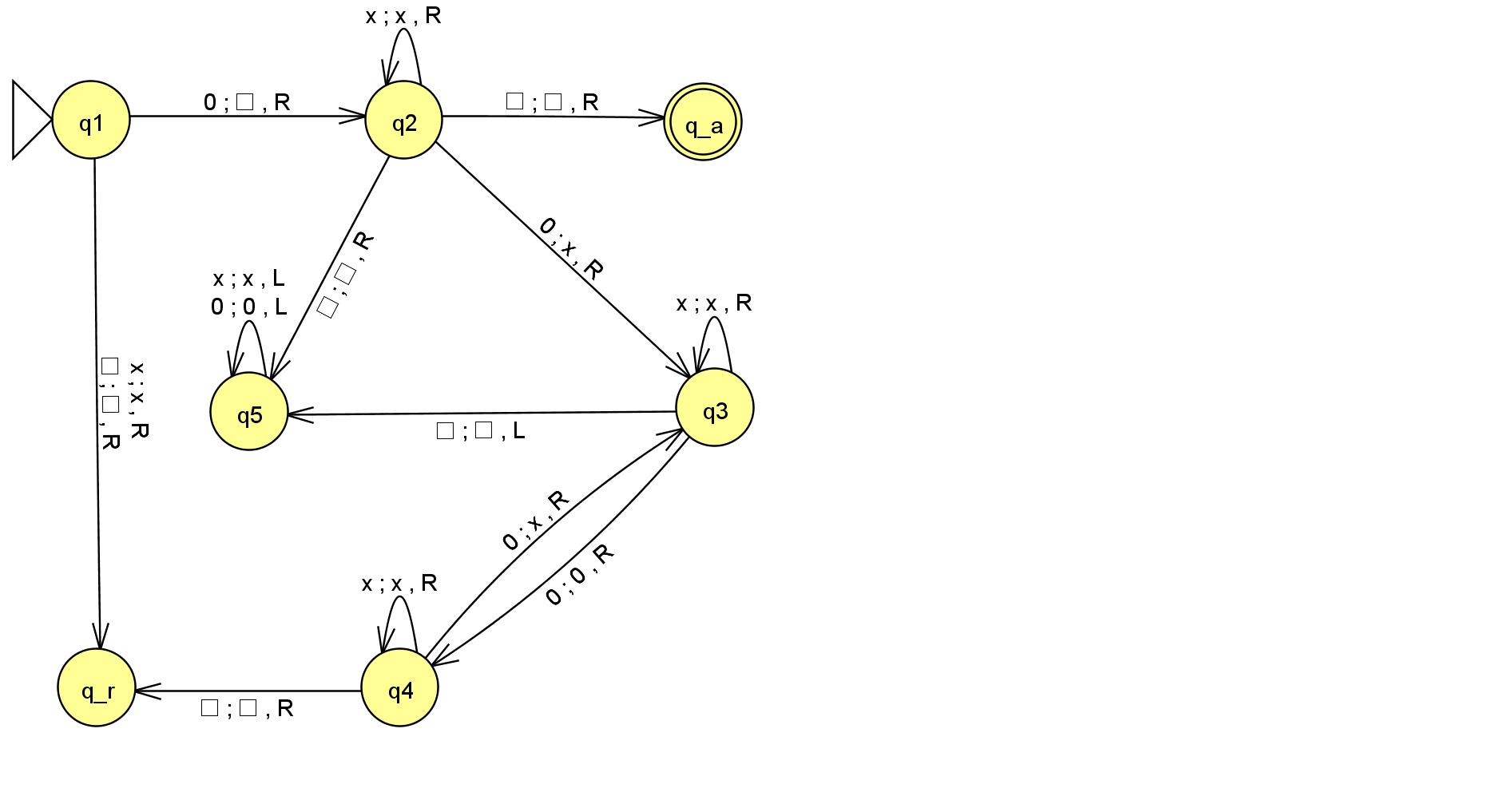
This exercise concerns the TM of Example 3.7 that decides , the language consisting of all strings of s whose length is a power of .

“On input string :

1. Sweep left to right across the tape, crossing off every other .
2. If in stage 1 the tape contained a single , accept.
3. If in stage 1 the tape contained more than a single and the number of s was odd, reject.
4. Return the head to the left-hand end of the tape.
5. Go to stage 1.”

The formal description of is:

* ,
* , and
* .
* is described by the state diagram below.[[1]](#footnote-2)
* The start, accept, and reject states are , , and , respectively.



Give the sequence of configurations that enters when started on .

#### Solution

### practice (sipser Exercise 3.1d)

Give the sequence of configurations that enters when started on .

### Practice (Sipser Exercise 3.5a)

Examine Sipser’s formal definition of a Turing machine to answer the following questions, and explain your reasoning.

1. Can a TM ever write the blank symbol ␣ on its tape?
2. Can the tape alphabet be the same as the input alphabet ?
3. Can a TM’s head ever be in the same location in two successive steps?
4. Can a TM contain just a single state?

#### Solution

1. Yes. The blank symbol ␣ is by definition an element of the tape alphabet .
2. No. The blank symbol ␣ is by definition not an element of the input alphabet .
3. No. According to Sipser’s definition, the tape head always moves left or moves right.
4. No. According to Sipser’s definition, , , and .

### practice (Sipser exercise 3.8b)

Give an implementation level description of a TM that decides the language

#### Solution

1. In the JFLAP image, represents ␣, represents , and represents . [↑](#footnote-ref-2)