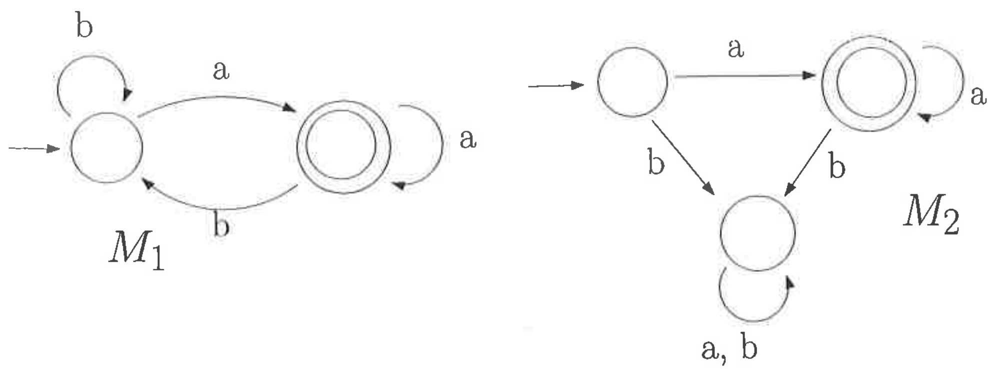

Figure 1: A finite automaton M_1 (on left) and a finite automaton M_2 (on right).

1. Define $A_{DFA} = \{ < B, w > \mid B \text{ is a DFA that accepts input string } w\}$,

$$\text{INCL}_{DFA} = \{ < B, C > \mid B, C \text{ are DFAs and } L(B) \subseteq L(C) \}.$$

Recall that DFA stands for "deterministic finite automaton". Answer the following questions (for M_1 and M_2 given in the above figure) and give reasons for your asnwers.

(a) Is $< M_1, baab > \in A_{DFA}$? No. M_1 does not accept baab.

(b) Is $< M_1, abba > \in A_{DFA}$? Yes. M_1 accepts abba.

(c) Is $< M_2, aa > \in A_{DFA}$? Yes. M_2 accepts aa.

(d) Is $< M_2, aaba > \in A_{DFA}$? No. M_2 does not accept aaba.

(e) Is $< aaa > \in A_{DFA}$? No. Incorrect input format.

(f) Is $< M_1, M_2 > \in \text{INCL}_{DFA}$? No. $ba \in L(M_1) - L(M_2)$.

(g) Is $< M_2, M_1 > \in \text{INCL}_{DFA}$? Yes. $\underbrace{L(M_2)}_{= aa^*} \subseteq L(M_1)$.

(h) Is $< M_2, M_2 > \in \text{INCL}_{DFA}$? Yes. $L(M_2) \subseteq L(M_2)$.

2. Give an implementation-level description of a **deterministic** one tape Turing machine M that decides the language over the alphabet $\Sigma = \{b, c\}$ consisting of all strings whose length is a power of two. That is, M decides the language:

$$A_{\text{power}} = \{ \ w \in \Sigma^* \mid \ \text{the length of } w \text{ is a power of two} \ \}.$$

Note: The first few powers of two are: 1, 2, 4, 8, 16, 32, 64, 128, ...

Decider for A_{power}:

Q = "On input $w \in \Sigma^*$

1. Scan the tape and mark every second symbol.
2. If in stage 1 the tape contained exactly one unmarked symbol, accept.
3. If in stage 1 the tape contained x unmarked symbols where $x > 1$ is odd, then reject.
4. Return tape head to left end of tape and continue from 1.

3. (a) Consider the language

$$A = \{ \; < M > \; | \; M \text{ is a Turing machine, and, } \underline{\text{no}} \text{ string of } L(M) \text{ contains string 00010 as a substring } \}.$$

Is the language A decidable or undecidable? Justify your answer.

A is undecidable. Two possible answers (one is sufficient)

i) A is a nontrivial semantic property and undecidable by Rice's theorem.

ii) Reduce A_{TM} to A. Suppose TM Q decides A. Then construct decider for A_{TM}:

P = "On input $\langle M, w \rangle$

1. Construct TM M_w = "On input x: 1. Run M on w
   2. If M accepts, accept.
   If M rejects, reject"
2. Run Q on input $\langle M_w \rangle$. If Q accepts, reject. If Q rejects, accept"

Note: $\langle M, w \rangle \in A_{TM}$ implies $L(M_w) = \Sigma^*$

$\langle M, w \rangle \notin A_{TM}$ implies $L(M_w) = \emptyset$

(b) Consider the language

$$B = \{ < M > \; | \; M \text{ is a Turing machine and there exists a DFA } C \text{ such that } L(M) \subseteq L(C) \}.$$

Recall that DFA stands for "deterministic finite automaton". Is the language B decidable or undecidable? Justify your answer.

B is decidable. Since set of all strings Σ^* is regular, we note that

$$B = \{ \langle M \rangle \mid M \text{ is a Turing machine} \}$$

A decider for B just needs to check that the input is a correct encoding of a TM.

4. (a) (3 marks) Consider the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = (2 \cdot n) - 1$. (Here $\mathbb{N} = \{1, 2, 3, \ldots\}$ is the set of positive integers.) Answer the following questions and give reasons for your answers.

   i. Is f onto? No. 2 is not an image of any number.

   ii. Is f one-to-one? Yes. $2x-1 = 2y-1$ implies $x=y$.

   iii. Is f a correspondence? No because it is not onto.

   (b) (3 marks) Are the following sets countable? For each case circle the correct answer – no explanation needed. If you circle both YES and NO, it is considered a wrong answer.

   - The set $\{0,1\}^*$. YES NO
   - The set of all subsets of $\{0,1\}^*$. YES NO
   - The set of all <u>finite</u> subsets of $\{0,1\}^*$. YES NO

   (c) (4 marks) Let $T = \{ (i,k) \mid i,k \in \mathbb{N} \}$. Show that T is countable.

The following list contains all elements of T:

Stage 1: list pairs (i,k) where $i+k=2$

Stage 2: list pairs (i,k) where $i+k=3$

⋮

Stage ℓ: list pairs (i,k) where $i+k=\ell+1$

⋮

5. Let $\Sigma = \{b, c\}$ and define

$$\text{ODD}_{\text{TM}} = \{ <M> \mid M \text{ is a deterministic Turing machine and } L(M) \text{ consists of all strings over } \Sigma \text{ having odd length} \}$$

Without using Rice's theorem show that ODD_{TM} is *undecidable*.

We reduce A_{TM} to ODD_{TM}. Suppose TM Q decides ODD_{TM}.
Then following TM decides A_{TM}:

P = "On input ⟨M,w⟩
1. Construct TM R:
   R = "On input x
   1. If x has odd length, accept.
   2. Run M on w. If M accepts, accept. If M rejects, reject".
2. Run Q on ⟨R⟩. If Q accepts, reject.
   If Q rejects, accept"

This works because:

$L(R) = \Sigma^*$ if M accepts w

$L(R) = (\Sigma^2)^*\Sigma$ if M does not accept w.