Transducers

Session 1: Mealy Machines

Aliaume LOPEZ
TA mail[*] Course page[†] Exercises page[‡]

March 11, 2024

1 Mealy Machines

Exercise 1 (True or False?). For each of the following functions, decide whether they can be realized by a Mealy Machine. In positive cases, provide the Mealy Machine, in negative cases, provide a proof that it cannot be realized.

- ☐ The function lowercase: $\Sigma^* \to \Sigma^*$, where Σ is the latin alphabet, that maps a word w to its lowercase variant. For instance, lowercase$(aAbcDA) = aabcda$.
- ☐ The function expandtabs: $\Sigma^* \to \Sigma^*$ that works on the alphabet Σ of ASCII characters, and replaces the *tab* codepoint `\t` by four spaces codepoints.
- ☐ The function $w \mapsto c^{|w|_a}$.
- ☐ The function sort: $\Sigma^* \to \Sigma^*$ that sorts its input, where Σ is a finite alphabet equipped with a total ordering $\leq$.
- ☐ The function $\Delta\colon \Sigma^* \to \Sigma^*$ that maps u to uu.
- ☐ The function swap: $\Sigma^* \to \Sigma^*$ that maps au to ua and ε to ε.
- ☐ The function swap$_2$: $\Sigma^* \to \Sigma^*$ that maps ua to au and ε to ε.

What are the extensions of Mealy Machines for which the above functions are computable? Among the above functions, which ones are continuous for the regular topology?

Exercise 2 (Arithmetic Circuits). The goal of this exercise is to prove that operations on binary numbers are possible. To that end we have to provide an encoding of tuples numbers, which we do as follows: a tuple $(n_1, \ldots, n_k) \in \mathbb{N}^k$ is represented on the alphabet $\{0, 1\}^k$ by writing the numbers in binary, and padding them with zeros so that the length matches. There are four variants of this encoding, obtained by deciding whether to pad on the left or the right, and whether to write numbers with the most significant bit on the left or the right.

1. For each of the four possible encodings, decide whether the map $(+)\colon \mathbb{N}^2 \to \mathbb{N}$ can be represented using a Mealy Machine.
2. For each of the four possible encodings, decide whether the map $(/3)\colon \mathbb{N} \to \mathbb{N}$ can be represented using a Mealy Machine.
3. Write a Mealy Machine that computes $(n, 4n)$ in binary.

[*] `ad.lopez@uw.edu.pl`
[†] `https://www.mimuw.edu.pl/~bojan/2023-2024/przeksztalcenia-automatowe-transducers`
[‡] `https://aliaumel.github.io/transducer-exercices/`

4. Deduce a Mealy Machine that computes $5n$ in binary by using the wreath product construction and the construction of the addition.

Exercise 3 (Bonus: Presburger Arithmetic). Prove that Presburger Arithmetic is decidable.
▷ Hint 1
▷ Hint 2

Exercise 4 (Flip Flop Machines). Prove that every flip-flop machine can be obtained by composing binary flip-flop machines. What is the number of intermediate machines that are needed?
▷ Hint 3
▷ Hint 4
▷ Hint 5

Exercise 5 (Regularity of Mealy Machines). The goal of this exercise is to understand the relationship between Mealy Machines and regular languages. Let $f : \Sigma^* \to \Gamma^*$ be a function computed by a Mealy Machine.

1. Prove that the image of Σ^* through f is a regular language.
2. Prove that the pre-image of Γ^* through f is a regular language.
3. Let L be a regular language, prove that $f(L)$ and $f^{-1}(L)$ are regular languages, i.e., that f is open and continuous for the regular topology.
4. Is every open and continuous map computable by a Mealy Machine?
5. A function $f : \Sigma^* \to \Gamma^*$ is Lipschitz for the prefix distance. What is the value of the Lipschitz constant?
6. The graph of a function $f : X \to Y$ is the subset $\mathsf{graph}(f) \subseteq X \times Y$ defined by $\{(x, y) \in X \times Y \mid f(x) = y\}$. Can you provide a necessary and sufficient condition on the graph of f for it to be representable using a Mealy Machine?

Exercise 6 (Decidability Properties of Mealy Machines). In this exercise, the goal is to understand what is decidable about Mealy Machines. For each of the following questions, prove (or disprove) that it is decidable, and in case of decidability, provide a precise complexity class.

1. Can we decide if two Mealy Machines compute the same function?
2. Can we decide if a Mealy Machine is surjective?
3. Can we decide if $f(w) \sqsubseteq g(w)$ for all $w \in \Sigma^*$?
4. Can we decide if a Mealy Machine is injective?
5. Can we decide if there exists $w \in \Sigma^*$ such that $f(w) = g(w)$?
6. Can we decide if a Turing machine computes a function that can be computed by a Mealy Machine?

▷ Hint 6

Exercise 7 (Variations on Mealy Machines). Describe the relationship between the expressiveness of the following variations of Mealy Machines:

1. Mealy Machines
2. Mealy Machines with lookaheads.
3. Sequential functions.

4. Mealy Machines with lookaheads with an ambiguous transition relation, but such that every run produces the same output.
5. Mealy Machines with lookaheads with an ambiguous transition relation, where the semantic is undefined if there are multiple runs producing different outputs.
6. Mealy Machines with transitions labelled by regular expressions.

Exercise 8 (Efficient String Matching)**.** The goal of this homework is to study the problem of string matching. That is, given a pattern $m \in \Sigma^*$ and a text $t \in \Sigma^*$, one wants to produce a text $m(t) \in (\Sigma \uplus \bar{\Sigma})^*$ where occurrences of m are overlined. To avoid ambiguity, we will overline non-overlapping occurrences of the pattern, starting from the left of the text t.

1. Is the function that underlines the starts of the matches computable by a Mealy Machine? By a sequential function? By a Mealy Machine with lookaheads?
2. Same question with underlining the ends of the matches.
3. Same question for the function m.
4. Conclude by providing an efficient algorithm to perform string matching. What is the (time/space) complexity in $|m|$? What is the (time/space) complexity in $|t|$?

2 Homework

Exercise 9 (Continuous Functions)**.** Prove that there exists uncountably many continuous functions from Σ^* to Γ^* for the regular topology. It is true for continuous and prefix preserving functions?
▷ Hint 7
▷ Hint 8
▷ Solution 1 (Complete Solution)

Exercise 10 (Stability properties of Sequential Functions)**.** We say that a function preserves prefixes if for all $u, v \in \Sigma^*$, $u \sqsubseteq_{\text{prefix}} v$ implies $f(u) \sqsubseteq_{\text{prefix}} f(v)$. Prove that the following propositions are equivalent for a function $f\colon \Sigma^* \to \Gamma^*$:

1. f is sequential.
2. f is continuous for the regular topology, Lipschitz for the prefix distance, and preserves prefixes.

2.0.1 Solution of the easy implication

Let f be a sequential function, computed by a sequential transducer $T = (Q_T, \delta_T, q_0^T, \lambda_T)$. Our first objective is to prove that f is continuous. To that end, let L be a regular language recognized by a finite monoid M, a morphism $\mu\colon \Gamma^* \to M$, and an accepting part $P \subseteq M$. We want to prove that $f^{-1}(L)$ is a regular language.

To that end, let us define $N = (Q_T \to Q_T) \times (Q_T \to M)$, with the multiplication $(\delta, \lambda) \cdot (\delta', \lambda') := (\delta \circ \delta', q \mapsto \lambda(q) \cdot \lambda'(q)))$. This is a finite monoid, where the pair $(\mathsf{id}, \mathsf{const}_1)$ is the identity element. Let us define $\varphi(a) := (\delta_T(a, \cdot), \lambda_T(a, \cdot))$ for all $a \in \Sigma$. It defines a morphism from Σ^* to N. Now, let us consider $S := \{(\delta, \lambda) \in N \mid \lambda(\delta(q_0)) \in P\}$. It is an easy check that $\varphi^{-1}(S) = f^{-1}(L)$, which proves that the latter is a regular language.

Let us now prove that f is Lipschitz for the prefix distance. Let us consider $K := \max\{|\lambda_T(a, q)| \mid a \in \Sigma, q \in Q_T\}$. It is an easy check that f is Lipschitz with constant K.

Finally, let us prove that f preserves prefixes. Let $u, v \in \Sigma^*$ be such that $u \sqsubseteq_{\text{prefix}} v$. Because (Q_T, q_0^T, δ_T) is a deterministic automaton, $f(v) = f(u) \cdot \lambda_T(\delta(q_0^T, u), u^{-1}v)$.

2.0.2 Solution of the hard implication

3 Cheat Sheet

3.1 Machines

Definition 1 (Mealy Machine). Let Σ and Γ be two alphabets. A Mealy Machine $\mathcal{M}$ is a tuple $(q_0, Q, \delta, \lambda)$ such that

1. Q is a finite set of states.
2. $q_0 \in Q$ is the initial state.
3. $\delta \colon Q \times \Sigma \to Q$ is a transition function.
4. $\lambda \colon Q \times \Sigma \to \Gamma$ is an output function.

The semantics of a Mealy Machine is given by the following inductive equations:

$$\mathcal{M}(w) := \mathcal{M}(q_0, w) \quad \mathcal{M}(q, \varepsilon) := \varepsilon \quad \mathcal{M}(q, au) := \lambda(q, a) \cdot \mathcal{M}(\delta(q, a), u)$$

Definition 2 (Flip-Flop Machine). A flip-flop machine is a Mealy Machine such that for all letters $a \in \Sigma$, either $\delta(\cdot, a)$ is the identity function, or it is a constant function. It is a binary flip-flop machine when $Q = \{0, 1\}$.

Definition 3 (Mealy Machine With Lookahead). Let Σ and Γ be two alphabets. A Mealy Machine with Lookahead $\mathcal{M}$ is a tuple $(q_0, Q, \delta, \lambda)$ such that

1. Q is a finite set of states.
2. $q_0 \in Q$ is the initial state.
3. $\delta \subseteq Q \times \Sigma \times Q$ is a transition relation.
4. $\lambda \colon Q \times \Sigma \times Q \to \Gamma$ is an output function.

In addition to this syntactic definition, we furthermore assume that for each $w \in \Sigma^*$, there exists at most one path in the automaton (q_0, Q, δ) starting from q_0 and reading w.

The semantics of the Mealy Machine is given by considering potential runs of the machine. Because of the absence of ambiguity, it defines a partial map $\mathcal{M} \colon \Sigma^* \rightharpoonup \Gamma^*$.

Definition 4 (Sequential Functions). Let Σ and Γ be two alphabets. A sequential transducer A is a tuple $(q_0, Q, \delta, \lambda)$ such that

1. Q is a finite set of states.
2. $q_0 \in Q$ is the initial state.
3. $\delta \colon Q \times \Sigma \rightharpoonup Q$ is a **partial** transition function.
4. $\lambda \colon Q \times \Sigma \to \Gamma^*$ is an output function.

The semantics is defined as for Mealy Machines.
Warning: this is sometimes called pure sequential functions.

3.2 Maths

Definition 5 (Presburger Arithmetic)**.** Formulas of the Presburger Arithmetic are built from the following grammar:

$$\varphi := \top \mid \bot \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid x = y + z$$

Given a valuation $\nu\colon \vec{x} \to \mathbb{N}$, we define the semantics of φ inductively as follows:

$$\begin{aligned}
\nu \models \top &\iff \text{true} \\
\nu \models \bot &\iff \text{false} \\
\nu \models \varphi \wedge \psi &\iff \nu \models \varphi \text{ and } \nu \models \psi \\
\nu \models \varphi \vee \psi &\iff \nu \models \varphi \text{ or } \nu \models \psi \\
\nu \models \neg\varphi &\iff \text{not } (\nu \models \varphi) \\
\nu \models \exists x.\varphi &\iff \text{there exists } n \in \mathbb{N} \text{ s.t. } \nu[x \mapsto n] \models \varphi \\
\nu \models x = y + z &\iff \nu(x) = \nu(y) + \nu(z)
\end{aligned}$$

Definition 6 (Topology and Continuous functions)**.** Let X be a set. A topology over X is a subset τ of $\mathcal{P}(X)$ closed under finite intersections and arbitrary unions. In a topological space (X, τ), the subsets in τ are called open subsets, and their complement are called closed subsets.

A function $f\colon (X, \tau) \to (Y, \theta)$ is continuous whenever for all open subset $U \in \theta$, its pre-image $f^{-1}(U)$ is an open subset of τ. Equivalently, it is continuous if the pre-image of closed subsets are closed subsets.

Definition 7 (Lipschitz functions)**.** A function $f\colon (X, d_X) \to (Y, d_Y)$ is Lipschitz if there exists a constant $K \geq 0$ such that for all $x_1, x_2 \in X^2$, $d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$.

Definition 8 (Prefix Distance)**.** Let Σ^* be a finite alphabet. The prefix distance between two words u, v is $|u| + |v| - 2|w|$ where w is the longest common prefix of u and v.

Definition 9 (Regular Topology)**.** Let Σ be a finite alphabet. We equip Σ^* with a metric distance as follows: to a pair of words u, w, we associate the minimal size $s(u, w)$ of a deterministic automaton that separates u from w. The distance between two words u and w, is defined as $d(u, w) := 2^{-s(u,w)}$. The regular topology is the topology defined by this metric on Σ^*.

Equivalently, the regular topology is the coarsest topology containing the regular languages as closed subsets.

A Hints

Hint 1 (Exercise 3 Encoding of numbers and formulas). Encode a formula $\varphi(\vec{x})$ as a **regular** language of $\mathbb{N}^X$, where numbers are encoded in binary with the most significant bit is on the left, and the padding is on the right.

Hint 2 (Exercise 3 Presburger Operators). Start by proving that each of these operations are computed by Mealy Machines.

1. The equality operator $(=)\colon \mathbb{N}^3 \to \{0,1\}$.
2. The addition operator $(+)\colon \mathbb{N}^2 \to \mathbb{N}$.
3. The existential quantifier $(\exists x)\colon \mathbb{N}^{x\vec{y}} \to \mathbb{N}^{\vec{y}}$.

Hint 3 (Exercise 4 Encode states). Given a state $q \in Q$, compute using a binary flip-flop machine the sequence of approximated states $\{q, \neg q\}$.

Hint 4 (Exercise 4 For the upper bound). Use a binary encoding to obtain a logarithmic number of intermediate machines.

Hint 5 (Exercise 4 For the lower bound). What can you say about a machine that shifts its input by one position?

Hint 6 (Exercise 6 Deciding Injectivity). Consider the set $\{(u,v) \in \Sigma^* \times \Sigma^* \mid f(u) = f(v)\}$, and show that it is a regular language.

Hint 7 (Exercise 9 The alphabet does not matter). Consider the set of all functions from $\mathbb{N}$ to $\mathbb{N}$.

Hint 8 (Exercise 9 Sufficient conditions for continuity). Show that the following conditions are sufficient for a function $f\colon \mathbb{N} \to \mathbb{N}$ to be continuous:

1. $\forall n \in \mathbb{N}, f(n)$ is a factorial,
2. $\liminf_{n\to\infty} f(n) = \infty$.

B Solutions

Solution 1 (Solution to Exercise 9)**.** We will follow the hints given, namely that the alphabet does not matter and that it is quite easy to be continuous. Indeed, consider $f\colon \mathbb{N} \to \mathbb{N}$ such that $f(n)$ is a factorial number, and $\liminf_{n\to\infty} f(n) = \infty$. Let $L \in \{1\}^* \simeq \mathbb{N}$ be a regular language, i.e., a language recognized by some finite monoid M, with a morphism $\mu\colon \mathbb{N} \to M$ and an accepting part $P \subseteq M$. Our goal is to prove that $f^{-1}(L)$ is a regular language too.

Let $n \in \mathbb{N}$ such that $f(n) \in L$, by definition it means that $\mu(f(n)) \in P \subseteq M$. Recall that for all $n \in \mathbb{N}$, there exists $m \in \mathbb{N}$ such that $f(n) = m!$, i.e., $f(n) = 1^{m!}$, as a consequence, $\mu(f(n)) = \mu(1)^{m!}$.

By standard results on finite monoids, there exists an idempotent e in M such that for all $m \geq |M|$, $\mu(1)^{m!} = e$. Let us sketch the proof for completeness purposes. This is obtained by remarking that the semigroup generated by $\mu(1)$ is finite, hence contains an idempotent e, obtained for some power $\mu(1)^k$ with $k \leq |M|$. Unicity is obtained by noticing that if $e_1 = \mu(1)^{k_1}$ and $e_2 = \mu(1)^{k_2}$, then $e_1 = e_1^{k_2} = \mu(1)^{k_1 \times k_2} = e_2^{k_1} = e_2$.

Now, using the fact that $\liminf_{n\to\infty} f(n) = \infty$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \geq |M|!$. As a consequence, for all $n \geq n_0$, $\mu(f(n)) = e$. If $e \in P$, this proves that $f^{-1}(L) = \{n \mid n \geq n_0\} \cup \{n < n_0 \mid f(n) \in L\}$, the latter is a regular language as the union of two regular languages. Otherwise, $e \notin P$, and we conclude that $f^{-1}(L) = \{n < n_0 \mid f(n) \in L\}$, which is also a regular language.

To conclude, let us remark that there are uncountably many functions satisfying the above conditions. This can be proven by considering the following injection of $\mathbb{N}^{\mathbb{N}}$ (which is well known to be uncountable) into such functions as follows: to a sequence $(p_n)_{n\in\mathbb{N}}$ of numbers, one can associate the function $f\colon n \mapsto (\sum_{i=0}^{n} p_i)!$.

Remark that in the above injection, all functions are prefix preserving, hence the answer to the second question is also positive. This has to be compared with the Theorem charaterizing sequential functions.