Transducers

Session 5: Linear Regular Functions
Version: v0.0.10

Aliaume LOPEZ
TA mail* Course page† Exercises page‡

June 24, 2024

1 Logic

Exercise 1 (MSO Workout). Provide MSO transductions realizing the following functions:

1. *reverse*
2. *sort*
3. *swap-first-last*
4. *duplicate*

2 Atomisation and Deatomisation

Exercise 2 (Deatomisation of Bimachines). Let $\mathbb{A}$ be an infinite set, and $\mathcal{S}$ be the set of permutations of $\mathbb{A}$. We define atomic bimachines with input alphabet $(\Sigma \cup \mathbb{A})$ and output alphabet $(\Gamma \cup \mathbb{A})$ via the existence of a finite monoid M and a morphism $\mu\colon (\Sigma \cup \mathbb{A})^* \to M$ such that $\mu(a) = 1_M$ for all $a \in \mathbb{A}$, together with a production function $\pi\colon M \times (\Sigma \cup \mathbb{A}) \times M \to (\Gamma \cup \mathbb{A})^*$, satisfying the following equivariance property (where $\sigma \in \mathcal{S}$ is lifted from permutations over $\mathbb{A}$ to an action over $(\Gamma \cup \mathbb{A})^*$ and $(\Sigma \cup \mathbb{A})^*$ in the natural way):

$$\forall a \in \mathbb{A}, \forall \sigma \in \mathcal{S}, \pi(m, \sigma(a), n) = \sigma(\pi(m, a, n)) \quad .$$

An atom coding function is a function $c\colon \mathbb{A} \hookrightarrow \langle 1^* \rangle$, i.e., that maps every atom to a unique word of the form $\langle 1^k \rangle$ for some $k \in \mathbb{N}$. We say that a function $f\colon (\mathbb{A} \cup \Sigma)^* \to (\mathbb{A} \cup \Gamma)^*$ is deatomisable if there exists a rational function $f^\dagger\colon (\Sigma \cup \{\langle, \rangle, 1\})^* to (\Gamma \cup \{\langle, \rangle, 1\})^*$ such that for all atomic codes $c\colon \mathbb{A} \hookrightarrow \langle 1^* \rangle$, the following commutes

$$c \circ f = f^\dagger \circ c \quad .$$

Prove that the following are equivalent:

1. A function f is deatomisable,
2. It is realisable by an atomic bimachine.

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

Exercise 3 (Atomic Bimachines). Prove that the following are not computable by atomic bimachines.

1. The reverse function.
2. The duplicate function.
3. The unzip function.

Conclude that those cannot be computed by rational functions.

3 Pumping Lemmas

Exercise 4 (Pumping Bimachines). Let f be computed by a bimachine. We extend the function f by considering $f(w_1[w]w_2)$ to be the word produced by the bimachine when reading w, under the context w_1 and w_2.

1. Prove that $f(w_1 w[w]^{X \times n!} w w_2)$ is of the form $\alpha\beta^X\gamma$ for some α, β, γ, where n is the number of states of the automata in the bimachine.
2. Conclude that reverse, duplicate and unzip are not computable by bimachines.

Exercise 5 (Pumping Sweeping Transducers). This exercise is based on the notion of sweeping transducers and their study done by Baschenis et al. [Bas+15] and Baschenis et al. [Bas+16].

Let f be computed by a sweeping transducer. Provide an appropriate pumping lemma for f. Use this pumping argument to prove that $\mathtt{map\text{-}reverse}$ is not computable using a sweeping transducer.

Exercise 6 (Pumping for sweeping 2DFTs). Provide a pumping lemma for sweeping 2DFTs. Conclude that $\mathtt{map\text{-}reverse}$ is not computable using a sweeping transducer.

Exercise 7 (Sweeping Minimization). We define the sweeping number of a sweeping transducer the maximal number of sweeps it performs on any input words.

1. Prove that the sweeping number of a sweeping transducer is finite
2. Does there exist an algorithm that, given a transducer T, computes its sweeping number?
3. Describe an algorithm that, given a sweeping transducer T and a number k with the promise that T can be realized by a sweeping transducer of sweeping number k, constructs a such a transducer.
4. Given a sweeping transducer T and a number k, is it decidable whether T is realized by a sweeping transducer with sweeping number at most k?

▷ Hint 1

4 Well quasi orderings

Exercise 8 (Well-Quasi-Ordered Image). Let f be a function from Σ^* to Γ^*. We say that f generates a well-quasi-order whenever $f(\Sigma^*)$ is well-quasi-ordered for the factor relation. We say that f generates a k-well-quasi-order whenever $f(\Sigma^*)$ endowed (freely) with k distinguishing colours (unary predicates) is a well-quasi-order. Finally, we say that f generates an ∞-well-quasi-order whenever it generates a k-well-quasi-order for all $k \in \mathbb{N}$.

1. Is it decidable whether the image of f is a well-quasi-ordering when f is computedb y a 2DFT?
2. What about a bimachine? What about a Mealy Machine?
3. Prove that it is decidable whether f generates an ∞-well-quasi-ordering.

References

[Bas+15] Félix Baschenis et al. "One-way Definability of Sweeping Transducer". In: 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2015). Ed. by Prahladh Harsha and G. Ramalingam. Vol. 45. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 178–191. ISBN: 978-3-939897-97-2. DOI: 10.4230/LIPIcs.FSTTCS.2015.178. URL: https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.FSTTCS.2015.178.

[Bas+16] Félix Baschenis et al. "Minimizing Resources of Sweeping and Streaming String Transducers". In: 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016). Ed. by Ioannis Chatzigiannakis et al. Vol. 55. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016, 114:1–114:14. ISBN: 978-3-95977-013-2. DOI: 10.4230/LIPIcs.ICALP.2016.114. URL: https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2016.114.

A Hints

Hint 1 (Exercise 7 Use effective continuity). Recall that if a function f is computed by a 2DFT, then it is continuous, and even more: effectively continuous.