Correction of the Transducer Homework 1

EXERCISE 2.1 (Continuity)

The Zip function is not continuous.

$$\boxed{\underset{\substack{\uparrow \\ \text{not regular}}}{zip^{-1}((ab)^*)} \cap \underset{\substack{\uparrow \\ \text{regular}}}{a^*\#b^*} = \underset{\substack{\uparrow \\ \text{not regular}}}{a^n\#b^n}}$$

not regular $\Leftarrow$ regular + not regular

EXERCISE 2.2 (Flip-Flop)

Let $M = (Q, \delta, \lambda, q_0)$ be a flip flop machine.
Assume without loss of generality that $Q = \{0,1\}^k$
define the machine

$M_i := (Q_i, \delta_i, \lambda_i, q_i^0)$ as follows for $1 \leq i \leq k$.

- $Q_i := \{0,1\}$
- $\lambda_i(q,a) := q$
- $\delta_i(q,a) := \begin{cases} q & \text{if } \delta_a \text{ is the identity map.} \\ \underbrace{\pi_i}_{\text{the ith projection.}}(q') & \text{if } \delta_a \text{ is the constant map equal to } q' \end{cases}$
- $q_i^0 = \pi_i(q_0)$

Now the parallel composition

$$M_1 \| M_2 \| \cdots \| M_k : \Sigma^* \to (T \times \{0,1\}^k)^*$$

can be obtained by composing the machines sequentially with suitable projections.
the homomorphism : $\lambda : \Sigma \times Q \to T$ allows to obtain

$$\forall w \in \Sigma^*, \quad M(w) = \lambda_0 (M_1 \| \cdots \| M_k)(w)$$

the latter is using $k+1$ compositions. Note that by merging the homomorphism we can obtain k compositions.

and $\boxed{k = \lceil \log_2 |Q| \rceil}$ ☒

Let us prove that this bound is tight.

let $\mathcal{R} = (\Sigma, \delta, \lambda, q_0)$ we fix $a \in \Sigma$

- $\delta_a(q) = a$
- $q_0 = a$
- $\lambda_a(q) = q$

① $\mathcal{R}$ has $|\Sigma|$ states. ② $\mathcal{R}$ cannot be realised by less than $|\Sigma|$ states.

③ the composition of k binary flip flops is computable by a 2^k states flip flop.

① is obvious

② Assume by contradiction that there exits $M' = (Q', \delta', \lambda', q_0')$ realising $\mathcal{R}$ with $|Q'| < |\Sigma|$.
then: $\lambda_a : Q' \to \Sigma$ is not surjective.
let $b \in \Sigma \setminus \lambda'_a(Q')$

but $M'(ba) = \lambda'_b(q_0) \cdot \lambda'(\delta'(q_0, b), a)$
$\quad \| $
$\mathcal{R}(ba) = ab$

implies

$b \in \lambda'_a(Q')$ which is absurd

③ let $M_1, \dots, M_k$ be binary flip flop machines with states Q_i, transitions δ_i, productions λ_i and initial state q_0^i

note that
$$M_1 : \Sigma^* \to \Sigma_1^*$$
$$M_2 : \Sigma_1^* \to \Sigma_2^*$$
$$\vdots$$
$$M_k : (\Sigma^{k-1})^* \to \Gamma^*$$

let us define $\mathcal{M} = (Q, \lambda, \delta, q_0)$ via

- $Q = Q_1 \times \cdots \times Q_k$ of size 2^k.
- $\lambda : \Sigma \times Q \to \Gamma$
$$(a, (q_1, \dots, q_k)) \mapsto \lambda_k(\lambda_{k-1}(\underset{\uparrow}{\dots} , q_{k-1}), q_k)$$
$$\lambda_1(a, q_1)$$
- $\delta : \Sigma \times Q \to Q$
$$(x, (q_1, \dots, q_k)) \mapsto (\delta_1(x, q_1), \delta_2(\lambda_1(x, q_1), q_2), \dots)$$

. We claim that
$$\forall w \in \Sigma^*, \ M(w) = M_k \circ \cdots \circ M_1 (w)$$

and leave the proof as an exercise. ☐

With ① + ② + ③ we conclude that R cannot be obtained by less than $\lceil \log_2 k \rceil$ binary flip flop machines

CQFD

EXERCISE 2.3 (Decide injectivity).

Let f be a rational function from Σ^* to T^*.
We construct the language $\mathcal{L}_f$ over the alphabet

$$\Delta := T \uplus \Sigma \times \{ (_1,)_1, (_2,)_2 \}$$

Words in $\mathcal{L}_f$ are of the form

$$w \; (^{a}_{1} \; u v w \; (^{a}_{2} \; x \;)^{a}_{1} \;)^{a}_{2} \quad \text{such that}$$

1) the word w is well-bracketed for $(_1,)_1$ and (independently) well-bracketed for $(_2,)_2$ without nesting: so $(^{a}_{1} \, (^{b}_{1} \,)^{b}_{1} \,)^{a}_{1}$ is forbidden.

2) the $(_1 \;)_1$ parentheses enclose factors of w that are produced by f on the run over the word written on top of the $(_1)_1$ parentheses.

3) similar to 2) but for $(_2)_2$ parentheses.

Conditions 1), 2) and 3) are regular.

1) because there is no nesting.

2) because we transform the NFA with output for f into an NFA that ignores $(^{x}_{2},)^{x}_{2}$ letters and replaces

$$q \xrightarrow{a/u} q' \quad \text{by} \quad q \xrightarrow{(^{a}_{1}} \boxed{\text{read } u} \xrightarrow{)^{a}_{1}} q'$$

3) similar to 2).

Now

$$\mathcal{L}_f \cap \left\{ \begin{array}{c} \text{the } l_2 \text{ word and the } l_1 \text{ word} \\ \text{are diferent} \end{array} \right\} = \emptyset$$

iff f is injective

Checking if the two words are equal is not doable by a CFG!
__But__ we can update our definition so that one position of each top word is selected / distinguished.
and use a pushdown automata to validate that the two selected letters are distinct and appear at the same moment in both words

The construction of the automatas is done in polynomial time and deciding emptiness is done in polynomial time too.
Hence deciding injectivity of rational functions is in PTIME.

EXERCISE 2.4 (Windowed Transducers)

① Let M be a Mealy machine. There exists a computable K such that M is K-windowed $\Leftrightarrow$ M is windowed.

② Let M be a Mealy machine and $K \in \mathbb{N}$. One can build a mealy machine $W(M,K)$ that is K-windowed and such that

$$W(M,K) \equiv M \quad \text{if } M \text{ is } K\text{-windowed}$$

Using ① and ② we can decide if a mealy machine is windowed by computing K (①) and checking the equivalence between M and $W(M,K)$.

Let us now prove ① and ②.

Proof of (1) let $\mathcal{A} = (Q, \delta, \lambda, q_0)$ we write

$(M, \circ)$ for the finite monoid generated by $(\delta_a : Q \to Q)_{a \in \Sigma}$.

- we write δ_w for the function $q \mapsto \delta^*(q, w)$.
- recall that δ_w is idempotent iff $\delta_w \circ \delta_w = \delta_w$.
- Claim : there exists a computable n_0 such that for all word w $(|w| \geqslant n_0)$ we can write $w = w_1 w_2 w_3$ with δ_{w_2} idempotent and $|w_1 w_3| < n_0$

proof sketch: use Ramsey ◁

- Let us assume that $\mathcal{A}$ is K-windowed for some $K \in \mathbb{N}$

by definition

$$(*) \quad \begin{array}{l} \forall u, w, v \in \Sigma^*, \ |v| \geqslant K \\ \forall a \in \Sigma \end{array} \quad \lambda(\delta^*(q_0, uv), a) = \lambda(\delta^*(q_0, wv), a)$$

Now let us prove that $\mathcal{A}$ is n_0-windowed.

let $u, w \in \Sigma^*$ and $|v| \geqslant n_0$.
$a \in \Sigma$

we have $v = v_1 v_2 v_3$, $|v_1 v_3| < n_0$ and δ_{v_2} idempotent

in particular $\forall n \in \mathbb{N}$.

$$\begin{aligned} \delta^*(q_0, u v_1 v_2^n v_3) &= \delta_{v_3} \delta_{v_2}^n \delta_{v_1} \delta_u (q_0) \quad \text{[and similarly for } w \text{]}. \\ &= \delta_{v_3 v_1 u} (q_0) \end{aligned}$$

But using $(*)$ with a large enough $n \in \mathbb{N}$ so that $n|v_2| \geqslant K$.

$$\begin{array}{ccc} \lambda(\delta^*(q_0, u v_1 v_2^n v_3), a) & = & \lambda(\delta^*(q_0, w v_1 v_2^n v_3), a) \\ \parallel & & \parallel \\ \lambda(\delta^*(q_0, u v); a) & & \lambda(\delta^*(q_0, w v); a) \end{array}$$

$\boxtimes$

Proof of (2) let $M = (Q, \delta, \lambda, q_0)$ and $K \in \mathbb{N}_{\geq 1}$

we write $W(M,K) = (Q', \delta', \lambda', q_0')$ with

$Q' := \Sigma^{\leq K}$ $\qquad q_0' := \varepsilon$

$\delta'(u,a) :=$ suffix of ua of size $\leq K$.

$\lambda'(u,a) := \lambda(\delta^*(q_0,u), a)$

We claim that if M is K-windowed then
$$M \equiv W(M,K).$$
we prove this claim by showing

$$\forall w \in \Sigma^+, \forall a \in \Sigma, \quad \lambda(\delta^*(q_0,w),a) = \lambda'(\delta'^*(q_0',w),a)$$

which is done by noticing that if $w = w_1 w_2$ $\quad |w_2| = K$

$$\delta'^*(q_0', w) = w_2.$$

and
$$\lambda'(\delta'^*(q_0',w),a) \overset{\text{def}}{=} \lambda(\delta^*(q_0,w_2),a) \quad \left(\begin{array}{l}\text{because}\\ M \text{ is } K\text{-}\\ \text{windowed}\end{array}\right)$$
$$\| $$
$$\lambda(\delta^*(q_0,w),a)$$

for words w of size less than K a similar argument holds.

$\boxtimes$

EXERCISE 3.1 (Semantically Functional).

$(\lambda : Q \times \Sigma \times Q \to T^*)$

Let $A = (Q, \delta, F, q_0, \lambda)$ be an NFA with outputs that is functional.

We endow Q with an arbitrary total ordering $\leq$.

① One can build a rational function $f : \Sigma^* \to (\Sigma \uplus Q)^*$ that outputs the lexicographically smallest run of A on the input word.

② there is a rational function $g : (\Sigma \uplus Q)^* \to T^*$ taking as input a valid run

$$q_0 \, a \, q_1 \, b \, q_2 \cdots$$

and producing $\lambda(q_0, a, q_1) \, \lambda(q_1, b, q_2) \cdots$

or inputs ε and produces $A(\varepsilon)$

Remark that $g \circ f = A$ and is a rational function as a composition.

It is clear how to build ② using a DFA with outputs.

Let us focus on ①.

We start by using a function $\text{pad}'_2 : aaba \mapsto \#a\#a\#b\#a\#$ that adds hashes around letters and is clearly a rational function.

Then we build an MSO relabeling from $(\Sigma \uplus \{\#\})^*$ to $(\Sigma \uplus Q)^*$ as follows:

- $\varphi_a(x) = a(x)$ for all $a \in \Sigma$.

- $\varphi_q(x) = \exists (X_p)_{p \in Q}$.
  1) (X_p) codes a valid run of A on the hashes $\#$.
  2) $x \in X_q$
  3) for all $(Y_p)_{p \in Q}$ valid run of A, (X_p) is lexicographically smaller than (Y_p)

Now: 1) is MSO definable as in class

2) is MSO definable too

3) Can be checked by asserting that either

- $\forall i. \bigwedge_{q \in Q} [i \in X_q \Leftrightarrow i \in Y_q]$ (equality)

- $\exists i. \left(\forall j < i, \bigvee_{q \leq q'} j \in X_q \wedge j \in Y_{q'} \right)$

$\wedge \bigvee_{q < q'} i \in X_q \wedge j \in Y_{q'}$ (strict inequality)

$\boxed{/\!/\!/}$

Please note that turning an NFA into a DFA may require an exponential blowup in the number of states. Hence any "polynomial time solution" to the above problem was not plausible.