Bachelor thesis
Computing Science &
Mathematics

Radboud University

Transducer Degrees

Author:
Steven Bronsveld
`steven.bronsveld@student.ru.nl`
s1020191

Supervisor/assessor:
Prof. Dr. J.H. Geuvers
`h.geuvers@cs.ru.nl`

Second assessor:
Prof. Dr. H. Zantema
`h.zantema@cs.ru.nl`

January 10, 2021

Abstract

We explore the structure of transducer degrees. This is a partial order on infinite sequences known as streams. One stream σ is considered to be greater than another stream τ if there exists a finite state transducer (FST) that transduces σ to τ. We show properties of this order and show what techniques were used to prove these properties. Our main focus is the fact that the degree of $\langle n^2 \rangle = \mathtt{1}\mathtt{0}^0\mathtt{1}\mathtt{0}^1\mathtt{1}\mathtt{0}^4\mathtt{1}\mathtt{0}^9\mathtt{1}\mathtt{0}^{16}\dots$ is an atom. This means that all transducts of $\langle n^2 \rangle$ can be transduced back to $\langle n^2 \rangle$ or to the zero stream $\mathbf{0}$.

Contents

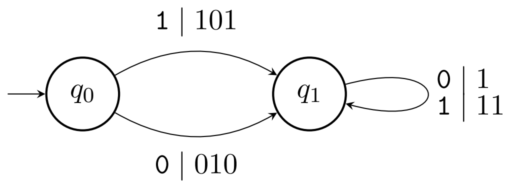
Chapter I

Introduction

Ordering streams

In this thesis, we explore the structure of *transducer degrees*. This is a largely unexplored topic in the field of theoretical computing science and automata theory. The concept was first introduced by *Jörg Endrullis* and his coworkers in the paper *Degrees of Streams* [1].

Transducer degrees are a way to order infinite sequences, also known as *streams*. Some streams can be transformed into other streams by a *finite state transducer* (FST). This is a kind of automaton that reads symbols from one stream and outputs symbols to a new stream. See the figure below.



An example of a FST

The state transitions can be read as "`input` | `output`". We will demonstrate how this FST produces an output stream from an input stream. In the beginning, the FST will be in state q_0, because this is the initial state (indicated by the arrow left of q_0). If we have a stream, for example, the alternating stream $A :=$ `10101010...`, this FST will read the first symbol (`1`) and output `101` and change the current state from q_0 to q_1. It will then read the second symbol (`0`), output `1` and change the state from q_1 to q_1 (so the state stays the same). Next, it will read the third symbol (`1`) and output `11`, and again keep in the same state. This process continues forever. The resulting output stream O looks like $O =$ `101111111...`.

Notice that we always read one symbol per transition, but can output multiple symbols at once. We call the act of transforming one stream into another using a FST *transduction*. Furthermore, the *transducts* of a stream σ are all streams that can be generated from σ by some FST. So the stream O above is one of the transducts of A.

We can order all streams using these transducers. We say that one stream σ is "bigger" than another stream τ if we can construct a FST that transduces σ to τ. Some streams can be transduced to each other. Whenever this is the case, we call the streams equivalent. Transducer degrees are the ordering of all streams where we treat equivalent streams as "the same". We call a set of equivalent streams a *transducer degree* and their collection *transducer degrees*.

Related work

These transducer degrees are similar to the well-known concept of *Turing degrees* also known as *degrees of unsolvability* [2, 3]. Their similarity is the topic of the paper *Degrees of Transducibility* [4]. This paper also compares *Mealy Degrees*, the degrees produced by *Mealy Machines*. These are finite state transducers that only output a single symbol per transition. This structure is the topic of [5].

Transducer degrees are created to gain more insight into the relation of streams. We know for example that the famous Thue-Morse stream can be transduced to the period doubling stream. The definition of these streams are given in Example 5.6 and a FST that can be used to transduce the Thue-Morse stream to the period doubling stream can be found in Example 6.3. It is argued that finite state transducers are a natural way to compare the complexity of different streams [1].

There are still a lot of open questions concerning transducer degrees as listed on the excellent website [6]. This website also gives a nice overview of papers related to this topic and the current state of the research. We highly encourage the reader to visit this site at:

`http://joerg.endrullis.de/research/finite-state-transducers/`

We will go over some of the open questions in the last chapter.

It is also possible to look at *permutation transducers*. These are similar machines to FSTs but with an additional requirement on the state transitions. In a FST, each state must have a transition that reads each input symbol. You cannot have a situation where you can get "stuck". A permutation transducer also requires that each state has incoming transitions for each input symbol. The example FST given above is not a permutation transducer, because the state q_0 does not have any incoming transitions. The properties of these permutation transducers have been studied in the paper *Classifying Non-periodic Sequences by Permutation Transducers* [7].

Research question

In this thesis we will be looking at the question:

What are the properties of transducer degrees?

To answer this question, we will delve into the work of *Jörg Endrullis*; the authority on this topic. We will explore the fine details of the proofs given in the papers written by him and his coworkers [1, 8, 9]. Some properties that we explore are:

- There exists a *bottom degree*.
- Every countable set of degrees has an *upper bound*.
- There are countably many *atom degrees*, these are degrees that only have the bottom degree below them.
- There are sets of degrees that do not have a *supremum*.

We fill focus mostly on the fact that a particular degree of streams is an atom namely, the degree of $\langle n^2 \rangle$.

Thesis overview

We will briefly go over the contents of each chapter.

Chapter III: Definitions We give some definitions that we will use throughout this thesis. We give examples and make the reader familiar with the notation.

Chapter IV: Initial investigation We prove some simple facts about transducer degrees. We solidify a few claims that were made in this introduction and prove a number of elementary facts.

Chapter V: Exploring transducts We look at the transducts of a certain class of streams, namely, those generated by spiralling functions. We work towards an elegant way to describe these transducts.

Chapter VI: Deep results We show some facts that are more difficult to prove. We will be using the results derived in Chapter V to show that the degree of the stream $\langle n^2 \rangle$ is an atom. We quickly go over the fact that the transducer degrees do not possess the supremum property.

Chapter II

Preliminaries

1 Prerequisites

We assume familiarity with basic automata theory. We will speak a lot about *finite state transducers*. These transducers are very similar to deterministic finite automata, as known in automata theory. We assume that the reader is familiar with mathematical proofs. Most of the proofs are not too difficult. However, the majority of this thesis consists of mathematics, so be warned.
Some of the mathematical concepts we assume the reader to be familiar are: *equivalence relations*, *partially ordered sets (posets)*, *the pigeonhole principle*, *the greatest common devisor (gcd)* and *the least common multiple (lcm)*.

2 About our proofs

In this thesis we will try to prove as many statements as possible, but we have to draw the line somewhere. When we require a FST for a proof, we will give the mathematical notation and in most cases a visual representation. If we want to be really precise, we would have to prove that a FST truly gives the required result by induction on the stream. We will not do this because this is quite tedious and does not give much additional insight.

3 Notation

We say that $0 \in \mathbb{N}$. Whenever we define an inline variable such as $n \geq 0$ or $x < n$, we assume this variable to be an integer ($\in \mathbb{Z}$) unless explicitly stated otherwise. Sometimes, when we define a new variable we denote this using the $:=$ sign to stress the fact that we *define* something and not *derive* something.

4 Citing

In this thesis, we heavily rely on the work of *Jörg Endrullis*. The primary focus of this thesis, the proof that the degree of squares is an atom, was shown in [8]. Whenever a definition, proposition, corollary, lemma or theorem is similar to one stated in another paper, we will link the source. All proofs that we give are of our own making, although they are often heavily inspired by their original versions. In many cases, we have introduced more examples, elaborated on parts that we thought were not clear, or introduced new concepts. We encourage the reader to look at the original papers.

Chapter III

Definitions

5 Streams

We start with a few definitions. We have intentionally made this section less mathematical, as we think this helps to get an intuitive sense of the notions introduced.

Definition 5.1. Words and streams.
A *stream* is a function $\sigma : \mathbb{N} \to \Sigma$, where Σ is a set of symbols, also called the *alphabet*. We define $\Sigma^{\mathbb{N}}$ to be the set of all streams over alphabet Σ. We call the stream that only contains zeros the *zero stream* denoted as $\mathbf{0} := (n \mapsto \mathtt{0})$. Because streams are functions, we say that $\sigma(n)$ is the symbol at index n for each $n \geq 0$. A *word* is a finite list of symbols. Let Σ^* be the set of words over the alphabet Σ. Let ϵ be the empty word. If $n \geq 0$, we denote a prefix of σ by $\sigma_{<n}$. This is the word of length n at the start of the stream. Similar definitions hold for $\sigma_{\leq n}, \sigma_{>n}$ and $\sigma_{\geq n}$. We can likewise define prefixes and suffixes of *words* instead of streams. We consider two streams, $\sigma, \tau \in \Sigma^{\mathbb{N}}$ to be the same if for all $n \geq 0$ we have $\sigma_{<n} = \tau_{<n}$.

For most of this thesis, we consider streams of two symbols, such that $\Sigma = \{0, 1\}$. We denote this by $\mathbf{2} := \{0, 1\}$. We will use $\mathbf{2}$ in the next definitions in favor of uniformity, but keep in mind that we could use any set of symbols Σ.

Definition 5.2. Word and stream operators.
Let $\sigma \in \mathbf{2}^{\mathbb{N}}$ be a stream $v, w \in \mathbf{2}^*$ words, $s \in \mathbf{2}$ a symbol and $k \geq 0$ an integer.

1. The *shift* operator.

$$\begin{aligned} \mathcal{S}^{\cdot}(.) &: \mathbb{N} \times \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}} \\ \mathcal{S}^k(\sigma) &:= n \mapsto \sigma(n+k) \end{aligned}$$

2. The infix *cons* operator that is overloaded to take streams and words as its second argument.

$$\begin{aligned} (:) &: \mathbf{2} \times \mathbf{2}^* \to \mathbf{2}^* \\ s : w &:= (s, w(0), w(1), ..., w(|w|-1)) \\ (:) &: \mathbf{2} \times \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}} \\ (s : \sigma)(n) &:= \begin{cases} s & \text{if } n = 0 \\ \sigma(n-1) & \text{if } n > 0 \end{cases} \end{aligned}$$

3. The infix *concatenation* operator defined by induction on the first argument. Note that this definition works for words and streams.

$$\begin{aligned}
(\cdot):\ & \mathbf{2}^* \times \mathbf{2}^* \to \mathbf{2}^* \\
& \epsilon \cdot v := v \\
& w \cdot v := w(0) : (w_{>0} \cdot v) && \text{if } |w| > 0 \\
(\cdot):\ & \mathbf{2}^* \times \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}} \\
& \epsilon \cdot v := v \\
& w \cdot v := w(0) : (w_{>0} \cdot v) && \text{if } |w| > 0
\end{aligned}$$

We introduce a *product* notation to make working with streams a lot more elegant.

Notation 5.3. Let $n, k \geq 0$ be integers, $u \in \mathbf{2}^*$ a word and $\vec{v} \in (\mathbf{2}^*)^n$ a list of words.

1. $u^k := \overbrace{u \cdot u \cdot ... \cdot u}^{k \text{ times}}$
2. $\prod_{i=0}^{k} v_i := v_0 \cdot v_1 \cdot ... \cdot v_{k-1}$
3. $v^\omega := v \cdot v^\omega$

(Note that the ω symbol in the superscript of Notation 5.3.3 is a lowercase omega)

In practice, the definitions of streams and words are quite intuitive. We show some examples below:

Example 5.4. Some words and streams.

$$\begin{aligned}
&\epsilon \in \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}^* && \mathtt{A} : \mathtt{BBA} = \mathtt{ABBA} \in \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}^* \\
&\mathtt{01010011} \in \mathbf{2}^* && \mathtt{1001} \cdot \mathtt{0110} = \mathtt{10010110} \in \mathbf{2}^* \\
&\mathtt{101010101...} \in \mathbf{2}^{\mathbb{N}} && \prod_{i=0}^{\infty} \mathtt{10}^i \in \mathbf{2}^{\mathbb{N}}
\end{aligned}$$

Although we will not focus on individual streams in this thesis, we will give some well-known streams to get familiar with the notation. These streams and their formal definitionswhere also mensioned in [1].

Definition 5.5. To give a formal definition of the streams below, we introduce the $szip_2$ function. Let $\sigma, \tau \in \mathbf{2}^{\mathbb{N}}$.

$$\begin{aligned}
&szip_2 : \mathbf{2}^{\mathbb{N}} \times \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}} \\
&\quad szip_2(s : \sigma, \tau) := s : szip_2(\tau, \sigma)
\end{aligned}$$

The $szip_2$ creates a new stream by alternating symbols of the two input streams. In Definition 10.3 we define a generalized version of $szip$.

Example 5.6 ([1])**.** Well-known streams.

- **Thue-Morse stream (A010060 [10])** This stream is obtained by starting with `0` and then repeatedly appending the boolean complement of the stream so far. We start with `0`, add the boolean complement of `0`, which is `1`, so we get `01`. We take the boolean complement of `01`, which is `10`, so we get `0110` and so on. We can give a formal definition. Note that this definition is different from the one given above. We encourage the reader to see that this other definition yields the same stream.

$$\begin{aligned} M &:= \mathtt{0} : X \\ X &:= \mathtt{1} : szip_2(X, Y) \\ Y &:= \mathtt{0} : szip_2(Y, X) \end{aligned}$$

$$M = \mathtt{0\ 1\ 10\ 1001\ 10010110\ 10010110011010...}$$

- **Period doubling stream (A096268 [10])** This stream is generated by starting with `01` and then repeatedly appending the stream so far to it, but taking the binary complement of the last symbol. So you start with `01` and then flip the last bit, so you get `00`, and append this. Continue this to infinity and you get the period doubling sequence. We give a formal definition below. Note that, this definition differs from the one described above.

$$\begin{aligned} PD &:= szip_2(\mathbf{0}, inv(PD)) \\ inv(\mathtt{1} : \sigma) &:= \mathtt{0} : \sigma \\ inv(\mathtt{0} : \sigma) &:= \mathtt{1} : \sigma \end{aligned}$$

$$PD = \mathtt{01\ 00\ 0101\ 01000100\ 01000101010001...}$$

- **Paperfolding stream (A014577 [10])** This stream is generated by starting with `1` and then inserting an alternating stream of `1` and `0` between the previous terms (including the start and end). So in the first iteration, you add `1` before the initial `1`, and `0` after it so you get `110`. The next iteration, you insert `1` in front, `0` between the first and the second symbol, `1` between the second and third symbol, and `0` at the end. So you get `1101100`. The formal definition below resembles this idea.

$$\begin{aligned} PF &:= szip_2(alt, PF) \\ alt &= \mathtt{1} \cdot (\mathtt{0} : alt) \end{aligned}$$

$$PF = \mathtt{11011001110010011101100011001 0}$$

6 Finite state transducers

Definition 6.1. Finite state transducer (FST).
A finite state transducer (FST) is a tuple $(Q, q_s, \delta, \lambda)$ where

- Q is a set of states.
- q_s is the initial state.
- $\delta : Q \times \mathbf{2} \to Q$ the transition function.
- $\lambda : Q \times \mathbf{2} \to \mathbf{2}^*$ the output function.

Definition 6.2. We extend the definition of the input function δ and output function λ of a FST $T := (Q, q_s, \delta, \lambda)$ such that we can give a word or a stream as input. Let $\sigma \in \mathbf{2}^{\mathbb{N}} \cup \mathbf{2}^*$ be a stream or a word.

$$\begin{aligned} \delta(q, \epsilon) &:= q & \qquad \delta(q, s : \sigma) &= \delta(\delta(q, s), \sigma) \\ \lambda(q, \epsilon) &:= q & \qquad \lambda(q, s : \sigma) &= \lambda(q, s) : \lambda(\delta(q, s), \sigma) \end{aligned}$$

We say that a transducer T *transduces* a stream $\sigma \in \mathbf{2}^{\mathbb{N}}$ to $\tau \in \mathbf{2}^{\mathbb{N}}$ if $\lambda(q_s, \sigma) = \tau$. We write $\sigma \geq_T \tau$ and $T(\sigma) = \tau$.

The best way to get familiar with this definition is by looking at an example.

Example 6.3. We define a finite state transducer $T := (Q, q_s, \delta, \lambda)$ where $Q := \{q_s, q_0, q_1\}$ is the set of states, q_s is the initial state and the transition and output functions δ, λ are defined by:

$$\begin{aligned} \delta(q_s, \mathtt{0}) &:= q_0 & \qquad \lambda(q_s, \mathtt{0}) &:= \epsilon \\ \delta(q_s, \mathtt{1}) &:= q_1 & \qquad \lambda(q_s, \mathtt{1}) &:= \epsilon \\ \delta(q_0, \mathtt{0}) &:= q_0 & \qquad \lambda(q_0, \mathtt{0}) &:= \mathtt{0} \\ \delta(q_0, \mathtt{1}) &:= q_1 & \qquad \lambda(q_0, \mathtt{1}) &:= \mathtt{1} \\ \delta(q_1, \mathtt{0}) &:= q_0 & \qquad \lambda(q_1, \mathtt{0}) &:= \mathtt{1} \\ \delta(q_1, \mathtt{1}) &:= q_1 & \qquad \lambda(q_1, \mathtt{1}) &:= \mathtt{0} \end{aligned}$$

This transducer is visualised below in Figure 6.4. This particular FST transduces the Thue-Morse stream (M) to the period doubling stream (PD) defined in Example 5.6

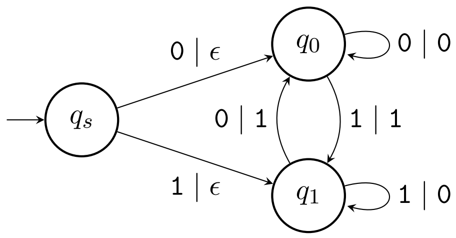


Figure 6.4: A FST that transduces the Thue-Morse stream (M) to the period doubling stream (PD).

The period doubling stream can be generated from the Thue-Morse stream by iterating through every symbol (starting with the second symbol), and checking if it is equal to the previous symbol. If so, output a `0`, otherwise ouput a `1`. One can consider state q_0 to be "the last symbol was a `0`" and q_1 "the last symbol was a `1`".

7 Transducer degrees

We can finally define the main topic of this thesis: *transducer degrees*. We start by defining an order on streams called "$\geq$".

Definition 7.1. For streams $\sigma, \tau \in \mathbf{2}^{\mathbb{N}}$ we say that $\sigma \geq \tau$ if and only if there exists a FST T such that $\sigma \geq_T \tau$. We say that $\sigma \equiv \tau$ if and only if $\sigma \geq \tau \wedge \tau \geq \sigma$.

This definition makes $(\mathbf{2}^{\mathbb{N}}, \geq)$ a partially ordered set, poset for short. We show this in Proposition 8.6. Furthermore, "$\equiv$" is an equivalence relation, as we show in Proposition 8.7. This makes it possible to speak about the equivalence classes of the quotient $(\mathbf{2}^{\mathbb{N}}/\equiv)$. We denote the equivalence class of a stream $\sigma \in \mathbf{2}^{\mathbb{N}}$ by $[\sigma] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$.

Definition 7.2. Transducer degree.
A *transducer degree* is an element of the set of equivalence classes $[\sigma] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ of the equivalence relation $\equiv$ defined above. We call this set of equivalence classes *transducer degrees*.

We can again define a partial order, this time on *transducer degrees*.

Definition 7.3. For transducer degrees $[\sigma], [\tau] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ we say that $[\sigma] \geq [\tau] \iff \sigma \geq \tau$. Note that we use the same symbol to denote the order on transducer degrees and streams.

We must show that this definition is well defined: that it does not depend on the specific representatives σ, τ. We must show that, for streams $\sigma, \sigma', \tau, \tau' \in \mathbf{2}^{\mathbb{N}}$, if $\sigma \equiv \sigma'$ and $\tau \equiv \tau'$, then $[\sigma] \geq [\tau] \iff [\sigma'] \geq [\tau']$. This follows directly from the definition of $\equiv$ and the fact that $\geq$ (as order on streams) is transitive.
In this thesis, we will mainly speak about facts of streams, but because of the above definition, we can directly translate this to facts about transducer degrees.

Chapter IV

Initial investigation

8 Basic statements

Now that we have the initial definitions out of the way, we can start with some basic properties of streams and transducers.

Proposition 8.1. Basic inequalities.
For any stream $\sigma \in \mathbf{2}^{\mathbb{N}}$ we have:

1. $\sigma \geq \mathbf{0}$.
2. $\sigma \geq \sigma$.
3. $\sigma \geq \mathcal{S}^k(\sigma)$ for any $k \geq 0$.
4. $\sigma \geq w \cdot \sigma$ for any $w \in \mathbf{2}^*$.

Proof.

1. The proof is given by a single FST that will transduce any stream $\sigma \in \mathbf{2}^{\mathbb{N}}$ to $\mathbf{0}$. Recall that $\mathbf{0} = \prod_{i=0}^{\infty} \mathtt{0} = \mathtt{0}^{\omega}$. Let $T := (\{q\}, q, \delta, \lambda)$ be a FST where

$$\begin{aligned} \delta(q, \mathtt{0}) &:= q \qquad & \lambda(q, \mathtt{0}) &:= \mathtt{0} \\ \delta(q, \mathtt{1}) &:= q \qquad & \lambda(q, \mathtt{1}) &:= \mathtt{0} \end{aligned}$$

   This transducer is shown in Figure 8.2.

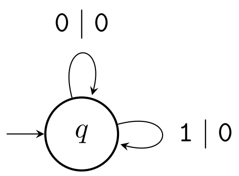


Figure 8.2: A FST that transduces any stream to $\mathbf{0}$.

   It is trivial that this transducer gives the required result.

2. We can create a FST that copies the input stream. Define the FST $T := (\{q\}, q, \delta, \lambda)$ where

$$\begin{aligned} \delta(q, \mathtt{0}) &:= q & \qquad \lambda(q, \mathtt{0}) &= \mathtt{0} \\ \delta(q, \mathtt{1}) &:= q & \qquad \lambda(q, \mathtt{1}) &= \mathtt{1} \end{aligned}$$

This transducer is shown in Figure 8.3. It is clear that this transducer provides us with the required result.

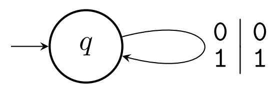


Figure 8.3: A transducer that copies the input stream.

3. Let $T := (Q, q_0, \delta, \lambda)$ be a FST where $Q := \{q_0, q_1, ..., q_{k-1}, q_k\}$ and

$$\begin{aligned} \delta(q_i, \mathtt{0}) &:= q_{i+1} & \qquad \lambda(q_i, \mathtt{0}) &:= \epsilon & \qquad &\text{for all } i < k \\ \delta(q_i, \mathtt{1}) &:= q_{i+1} & \qquad \lambda(q_i, \mathtt{1}) &:= \epsilon & \qquad &\text{for all } i < k \\ \delta(q_k, \mathtt{1}) &:= q_k & \qquad \lambda(q_k, \mathtt{1}) &:= \mathtt{1} \\ \delta(q_k, \mathtt{0}) &:= q_k & \qquad \lambda(q_k, \mathtt{0}) &:= \mathtt{0} \end{aligned}$$

This transducer is shown in Figure 8.4.

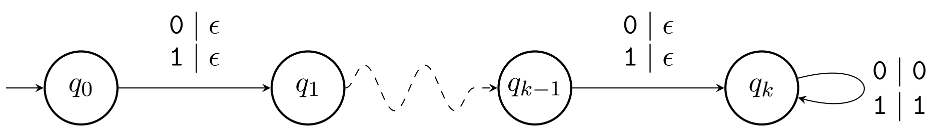


Figure 8.4: A transducer that shifts a stream by k symbols.

This transducer skips the first k symbols and then starts copying the stream.

4. Let $w \in \mathbf{2}^*$ be a word and $\sigma \in \mathbf{2}^{\mathbb{N}}$ a stream. We will show that $\sigma \geq_T w \cdot \sigma$ by the FST $T := (\{q_0, q_1\}, q_0, \delta, \lambda)$ where

$$\begin{aligned} \delta(q_0, \mathtt{0}) &:= q_1 & \qquad \lambda(q_0, \mathtt{0}) &:= w \cdot \mathtt{0} \\ \delta(q_0, \mathtt{1}) &:= q_1 & \qquad \lambda(q_0, \mathtt{1}) &:= w \cdot \mathtt{1} \\ \delta(q_1, \mathtt{0}) &:= q_1 & \qquad \lambda(q_1, \mathtt{0}) &:= \mathtt{0} \\ \delta(q_1, \mathtt{1}) &:= q_1 & \qquad \lambda(q_1, \mathtt{1}) &:= \mathtt{1} \end{aligned}$$

The FST T is shown in Figure 8.5.

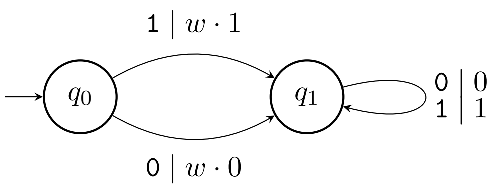


Figure 8.5: A transducer that prepends a word w.

$\square$

We will now prove that "$\geq$" forms a partial order and that "$\equiv$" is indeed an equivalence relation.

Proposition 8.6. The relation "$\geq$" is a partial order on the set of streams $\mathbf{2}^{\mathbb{N}}$.

Proof. We give a sketch of the proof. Let $\sigma, \tau, \kappa \in \mathbf{2}^{\mathbb{N}}$ be streams.

(reflexivity) $\sigma \geq \sigma$ follows directly from Proposition 8.1.2.

(antisymmetry) $\sigma \geq \tau \wedge \tau \geq \sigma \implies \tau \equiv \sigma$ as per the definition of "$\equiv$" given in Definition 7.1.

(transitivity) $\sigma \geq \tau \wedge \tau \geq \kappa \implies \sigma \geq \kappa$. Suppose $\sigma \geq \tau \wedge \tau \geq \kappa$. Then can find two transducers $T = (Q, q_s, \delta, \lambda)$ and $T' = (Q', q'_s, \delta', \lambda')$ such that $T(\sigma) = \tau \wedge T'(\tau) = \kappa$. We can now construct the *wreath product*: $\hat{T} := (Q \times Q', (q_s, q'_s), \delta, \lambda)$ where

$$\begin{array}{lll} \delta((q,q'),s) & := & (\delta(q,s),\ \delta'(q,s)) \qquad \text{for } s \in \mathbf{2} \\ \lambda((q,q'),s) & := & \lambda'(q',\ \lambda(q,s)) \qquad \text{for } s \in \mathbf{2} \end{array}$$

We claim that this construction gives the required result, namely that $\sigma \geq_{\hat{T}} \kappa$. We will not prove this.

$\square$

Proposition 8.7. The relation "$\equiv$" is an equivalence relation on the set of streams $\mathbf{2}^{\mathbb{N}}$.

Proof. Let $\sigma, \tau, \kappa \in \mathbf{2}^{\mathbb{N}}$ be streams.

(reflexivity) $\sigma \equiv \sigma$ follows directly from Definition 7.1 and the reflexivity of "$\geq$" as shown in Proposition 8.6.

$$\sigma \geq \sigma \wedge \sigma \geq \sigma \implies \sigma \equiv \sigma$$

(symmetry) $\sigma \equiv \tau \implies \tau \equiv \sigma$ follows directly from Definition 7.1:

$$\sigma \equiv \tau \implies \sigma \geq \tau \wedge \tau \geq \sigma \implies \tau \geq \sigma \wedge \sigma \geq \tau \implies \tau \equiv \sigma$$

(transitivity) $\sigma \equiv \tau \wedge \tau \equiv \kappa \implies \sigma \equiv \kappa$ follows from the transitivity of "$\geq$" as shown in Proposition 8.6.

$\square$

Theorem 8.8. There exists a bottom degree, namely:

$$[\mathbf{0}] = \{uv^{\omega} \mid u, v \in \mathbf{2}^*, |v| > 0\}$$

Proof. We give a sketch. The fact that $[\mathbf{0}]$ is a bottom degree follows from the fact that every stream can be transduced to $\mathbf{0}$ as shown in Proposition 8.1. We will not prove the fact that $[\mathbf{0}]$ is equal to the set of all *ultimately periodic streams.* $\square$

9 Function streams

In this section we define *function streams*, a way to transform a function $f : \mathbb{N} \to \mathbb{N}$ into a stream σ.

Definition 9.1. Function stream.
Let $f : \mathbb{N} \to \mathbb{N}$ be a function. We define the *function stream* $\langle f \rangle$ by:

$$\langle f \rangle := \prod_{i=0}^{\infty} \mathtt{1}\mathtt{0}^{f(i)} \tag{9.2}$$

We will call each sequence $\mathtt{1}\mathtt{0}^{f(i)}$ a *block*. Sometimes we will refer to function streams as block streams.

We can transform any function defined on the natural numbers into a stream. We will (ab)use the $\langle . \rangle$ notation to denote the following:

Notation 9.3. Overloading the $\langle . \rangle$ notation.

1. We will sometimes write $\langle f(n) \rangle$ instead of $\langle n \mapsto f(n) \rangle$. Examples:
   - $\langle n^2 \rangle = \langle n \mapsto n^2 \rangle$
   - $\langle 3n^3 + 2n + 1 \rangle = \langle n \mapsto 3n^3 + 2n + 1 \rangle$
2. We introduce the following set-like notation: $\langle f(n) \mid a \le n < b \rangle := \prod_{n=a}^{b-1} \mathtt{1}\mathtt{0}^{f(n)}$. Examples:
   - $\langle f(n) \mid 1 \le n < 4 \rangle$ for $f(n) := 2n$ gives $\langle f(n) \mid 1 \le n < 4 \rangle = \prod_{n=1}^{3} \mathtt{1}\mathtt{0}^{2n} = \mathtt{1}\mathtt{0}^2\mathtt{1}\mathtt{0}^4\mathtt{1}\mathtt{0}^6 = \mathtt{100100001000000}$
   - $\langle f(3+2n) \mid a \le n < b \rangle$ for $a, b \ge 0$ and $f : \mathbb{N} \to \mathbb{N}$ gives $\langle f(3+2n) \mid a \le n < b \rangle = \prod_{n=a}^{b} \mathtt{1}\mathtt{0}^{f(3+2n)}$

Definition 9.4. Shift function.
We define the *shift function* $\mathcal{S}^k : (\mathbb{N} \times (\mathbb{N} \to \mathbb{N})) \to (\mathbb{N} \to \mathbb{N})$ by $\mathcal{S}^k(f) := n \mapsto f(n+k)$ for all $k \ge 0$. As the name suggests, $\mathcal{S}^k$ shifts a given function k places, for example $\mathcal{S}^2(n \mapsto 2n) = n \mapsto 2n + 2$

Lemma 9.5. Simple facts about function streams.
Let $f : \mathbb{N} \mapsto \mathbb{N}$ be a natural function.

1. $\langle \mathcal{S}^a(f) \rangle = \langle n \mapsto f(n+a) \rangle \equiv \langle f \rangle$ for all $a \ge 0$.
2. $\langle af(n) \rangle \equiv \langle f(n) \rangle$ for all $a > 0$.
3. $\langle f(n) + a \rangle \equiv \langle f(n) \rangle$ for all $a \ge 0$.
4. $\langle f(n) \rangle \ge \langle f(an) \rangle$ for all $a > 0$.

Proof.
As per Definition 7.1, we will seperately show "$\ge$" and "$\le$" when we want to show an equivalence "$\equiv$".

1. $(\geq)$ We create a FST that skips the first k blocks of $\langle f \rangle$. Let $T := (Q, q_0, \delta, \lambda)$, where $Q := \{q_0, q_1, ..., q_{k+1}\}$. Define δ and λ as follows:

$$
\begin{aligned}
\delta(q_i, \mathtt{0}) &:= q_i & \lambda(q_i, \mathtt{0}) &:= \epsilon & &\text{for } i < k+1 \\
\delta(q_i, \mathtt{1}) &:= q_{i+1} & \lambda(q_i, \mathtt{1}) &:= \epsilon & &\text{for } i < k+1 \\
\delta(q_{k+1}, \mathtt{0}) &:= q_{k+1} & \lambda(q_{k+1}, \mathtt{0}) &:= \mathtt{0} \\
\delta(q_{k+1}, \mathtt{1}) &:= q_{k+1} & \lambda(q_{k+1}, \mathtt{1}) &:= \mathtt{1}
\end{aligned}
$$

This transducer is shown below in Figure 9.7.

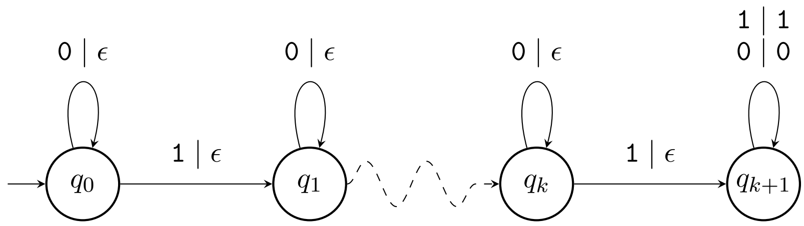


Figure 9.6: A transducer that shifts a function stream by k blocks.

$(\leq)$ Let $w := \langle f(n) \mid 0 \leq n < k+1 \rangle$ be the word equal to the first k blocks of $\langle f \rangle$. By Proposition 8.1.4 we know that $\langle \mathcal{S}^k(f) \rangle \geq w \cdot \langle \mathcal{S}^k(f) \rangle = \langle f \rangle$.

2. $(\geq)$ The following FST T replaces every factor of $\mathtt{0}^a$ by $\mathtt{0}$. Notice that we can assume $f(n)$ to be a multiple of a for each $n \geq 0$. Let $T := (Q, q_0, \delta, \lambda)$ with $Q := \{q_0, q_1, ..., q_a\}$ and the input and output functions defined by

$$
\begin{aligned}
\delta(q_i, \mathtt{0}) &:= q_{i+1} & \lambda(q_i, \mathtt{0}) &:= \epsilon & &\text{for } i < a \\
\delta(q_i, \mathtt{1}) &:= q_i & \lambda(q_i, \mathtt{1}) &:= \mathtt{1} & &\text{for } i < a \\
\delta(q_a, \mathtt{0}) &:= q_0 & \lambda(q_a, \mathtt{0}) &:= \mathtt{0} \\
\delta(q_a, \mathtt{1}) &:= q_0 & \lambda(q_a, \mathtt{1}) &:= \mathtt{1}
\end{aligned}
$$

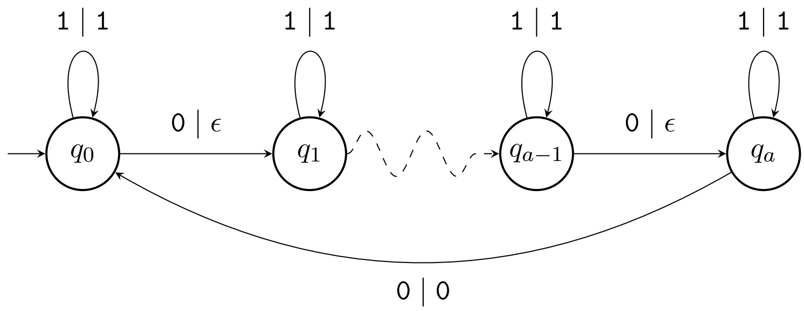


Figure 9.7: A transducer that replaces $\mathtt{0}^a$ by $\mathtt{0}$.

$(\leq)$ Let T be a FST that replaces each $\mathtt{0}$ by $\mathtt{0}^a$ defined by $T := (\{q_s\}, q_s, \delta, \lambda)$, where

$$
\begin{aligned}
\delta(q_s, \mathtt{0}) &:= q_s & \lambda(q_s, \mathtt{0}) &:= \mathtt{0}^a \\
\delta(q_s, \mathtt{1}) &:= q_s & \lambda(q_s, \mathtt{1}) &:= \mathtt{1}
\end{aligned}
$$

3. ($\geq$) This transduction is realised by a FST that removes a zeros from each block.
   ($\leq$) Adding a zeros to each block can also be done by a FST.

4. A FST can output every a-th block, and replace the rest by ϵ. Notice that the transduction back is not always possible because a FST cannot guess the missing values of an arbitrary function.

$\square$

10 Upper bounds

We will quickly go over the existence of upper bounds of a set of streams. An easy way to create an upper bound is to *zip* the set of streams. We will define two zipping functions, one that works on streams *szip* and one that works on functions: *zip*.

Definition 10.1. *zip*.
Let $k > 0$ and $f_0, f_1, ..., f_{k-1} : \mathbb{N} \to \mathbb{N}$ be natural functions.
Define $zip_k(f_0, f_1, ..., f_{k-1}) : \mathbb{N} \to \mathbb{N}$ by

$$zip_k(f_0, f_1, ..., f_{k-1})(kn + i) = f_i(n) \qquad \text{where } i < k, n \geq 0 \tag{10.2}$$

Definition 10.3. *szip* (short for stream-zip).
Let $k > 0$ and $\sigma_0, \sigma_1, ..., \sigma_{k-1} \in (\mathbf{2}^{\mathbb{N}})^k$ be natural streams.
Define $szip_k(\sigma_0, \sigma_1, ..., \sigma_{k-1}) \in \mathbf{2}^{\mathbb{N}}$ by

$$szip_k(\sigma_0, \sigma_1, ..., \sigma_{k-1}) := zip_k(\sigma_0, \sigma_1, ..., \sigma_{k-1}) \tag{10.4}$$

Although the two definitions are identical, we want to stress that we sometimes zip *streams* together and sometimes zip *other functions* together.

Example 10.5.

$$szip_2(\mathbf{0}, (1)^\omega) = (01)^\omega$$
$$zip_2(n \mapsto 2n, n \mapsto n^2) = \begin{cases} n \mapsto n & \text{if } n \text{ is even} \\ n \mapsto \lfloor \frac{n-1}{2} \rfloor^2 & \text{if } n \text{ is odd} \end{cases}$$

Definition 10.6. If $S \subseteq (\mathbf{2}^{\mathbb{N}}/\equiv)$ is a set of transducer degrees. We say that $[u] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ is an *upper bound* of S, if for all $[\sigma] \in S$ we have $[u] \geq [\sigma]$.

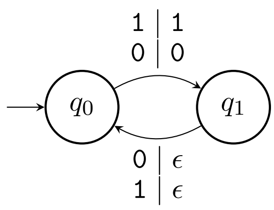
We can now highlight **an important difference** between the partial order on streams $(\mathbf{2}^{\mathbb{N}}, \geq)$ and the partial order on transducer degrees $(\mathbf{2}^{\mathbb{N}}/\equiv, \geq)$. It only makes sense to talk about upper bounds in the partial order of transducer degrees. Within the poset of streams, any two equivalent streams would be each others upper bound.

Lemma 10.7. If $\sigma, \tau \in \mathbf{2}^{\mathbb{N}}$, then $[szip_2(\sigma, \tau)]$ is an upper bound of $\{[\sigma], [\tau]\}$.

Proof. We show that $szip_2(\sigma, \tau) \geq_T \sigma$ and $szip_2(\sigma, \tau) \geq_{T'} \tau$ by giving FSTs T and T':

$$\begin{array}{lll} T := (\{q_0, q_1\}, q_0, \delta, \lambda) & T' := (\{q_0, q_1\}, q_1, \delta, \lambda) & \\ \delta(q_0, s) := q_1 & \lambda(q_0, s) := s & \text{for } s \in \mathbf{2} \\ \delta(q_1, s) := q_0 & \lambda(q_1, s) := \epsilon & \text{for } s \in \mathbf{2} \end{array}$$

See Figure 10.8 for a visualisation. Note that the only difference between T and T' is the initial state.



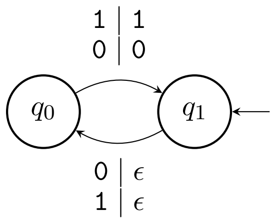


Figure 10.8: A FST that drops every second symbol (T, left) and a FST that drops every second symbol with an offset of one (T', right)

□

Lemma 10.9. If $f, g : \mathbb{N} \to \mathbb{N}$, then $[\langle zip_2(f,g)\rangle]$ is an upper bound of $\{[\langle f\rangle], [\langle g\rangle]\}$.

Proof. We need to show that $\langle zip_2(f,g)\rangle \geq \langle f\rangle$ and $\langle zip_2(f,g)\rangle \geq \langle g\rangle$. This is easily done by applying Lemma 9.5.

$$\langle n \mapsto zip_2(f,g)(n)\rangle \overset{9.5.4}{\geq} \langle n \mapsto zip_2(f,g)(2n)\rangle = \langle f\rangle$$
$$\langle n \mapsto zip_2(f,g)(n)\rangle \overset{9.5.1}{\equiv} \langle n \mapsto zip_2(f,g)(n+1)\rangle \overset{9.5.4}{\geq} \langle n \mapsto zip_2(f,g)(2n+1)\rangle = \langle g\rangle$$

□

Lemma 10.10 ([1])**.** Every countable set of degrees $\{[\sigma_i]\}_{i=0}^{\infty}$ has an upper bound.

Now that we have seen that upper bounds exist, we can wonder if two degrees have a *least upper bound* also known as the *supremum*.

Definition 10.11. A set $S \subseteq (\mathbf{2}^{\mathbb{N}}/\equiv)$ of transducer degrees has a *least upper bound* or a *supremum* $[s] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ if, for each upper bound $[u] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ of S we have $[u] \geq [s]$.

In Section 19 we will illustrate that suprema do not always exist.

11 Atoms

We have seen that all streams can be transduced to the zero stream $\mathbf{0}$. A logical next question is if there are *atom* degrees, that is, degrees that only have the zero degree $[\mathbf{0}]$ strictly below them.

Definition 11.1. A degree $[\sigma] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ is called an *atom* if $[\sigma] \geq [\tau]$ implies $[\tau] = [\sigma] \vee [\tau] = [\mathbf{0}]$. Alternatively, $[\sigma]$ is an atom if $\sigma \geq \tau \implies \tau \geq \mathbf{0} \vee \tau \geq \sigma$.

Theorem 11.2 ([1])**.** $[\langle n \rangle]$ is an atom.

We will not prove the above theorem here. We will focus on a bigger fish, namely the fact that $[\langle n^2 \rangle]$ is an atom. To prove this, we have quite a way to go. We start by introducing a new class of functions that will aid us in understanding transducts of function streams.

Chapter V

Exploring transducts

In this chapter, we develop techniques to better characterize the transducts of some streams. These techniques were shown in the paper *The degree of squares is an Atom* [8]. We will use these techniques to conclude some deeper results about transducer degrees in the next chapter. We will create an elegant form of the transducts of some streams, namely, those created by spiralling functions. We will do this in the following steps:

- We define the class of spiralling functions.
- We show that, when f is spiralling and σ is a transduct of $\langle f \rangle$ then:

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)}$$

- We show that this double product can be simplified to:

$$\sigma \equiv \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\varphi(i,j)}$$

- We introduce the concept of *mass products* and *weight displacements* to derive that:

$$\sigma \equiv \langle \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \rangle$$

The details of this derivation can be found in the next sections. It is good to keep in mind that we are working towards an elegant form to characterize the transducts of some streams.

12 Properties of spiralling functions

We define the class of *spiralling functions* and show some properties. We denote the set of spiralling functions by "$\circledcirc$". This class of functions is useful because we can say a lot about the transducts σ of $\langle f \rangle$ whenever f is spiralling.

Definition 12.1 ([8])**.** A function $f : \mathbb{N} \to \mathbb{N}$ is called *spiralling* ($f \in \circledcirc$) if and only if

(a) For all $m \geq 0$ there exists $N \geq 0$ such that

$$f(n) \geq m \qquad \text{(for all } n \geq N\text{)}$$

(b) For all $m > 0$ there exists $N \geq 0,\ p > 0$ such that

$$f(n+p) \equiv f(n) \bmod m \qquad \text{(for all } n \geq N\text{)}$$

This definition states that a function f is spiralling if $\lim_{n\to\infty} f(n) = \infty$ (12.1.a) and that, for any $m > 0$ the function $n \mapsto f(n) \bmod m$ is ultimately periodic (12.1.b). We give some examples of spiralling functions, and their straightforward proofs.

Proposition 12.2.

1. $Id_{\mathbb{N}} \in \circledcirc$
2. $(n \mapsto g^n) \in \circledcirc$ where $g > 1$

Proof.

1. (a) Let $m \geq 0$, take $N := m$, then for all $n \geq N$ we have

$$Id_{\mathbb{N}}(n) = n \geq N = m$$

   (b) Let $m > 0$, take $N := 0$ an $p := m$, then for all $n \geq N$ we have

$$Id_{\mathbb{N}}(n+m) = n + m \equiv n = Id_{\mathbb{N}}(n) \bmod m$$

2. (a) Let $m \geq 0$. Then for all $n \geq m$, we have $g^n \geq g^m \geq m$.

   (b) Let $m > 0$. By the pigeonhole principle, there exists $a, b > 0$ such that $g^{a+b} \equiv g^a \bmod m$, because there are more than m positive integers, so that g^{a+b} and g^a must be equivalent modulo m for some $a, b \geq 0$. We choose $p := b$ and $N := a$. For all $n \geq N$, we can write $n = c + a$ for some $c \geq 0$, such that we can derive:

$$g^{n+p} = g^{c+a+p} = g^c g^{a+b} \equiv g^c g^a = g^{c+a} = g^n \bmod m$$

□

We will now give closure properties of spiralling functions.

Theorem 12.3. Closure properties of $\circledcirc$.
Let $f, g \in \circledcirc$ and $c > 0$. Then we have:

1. $\mathcal{S}^c(f) \in \circledcirc$
2. $c + f \in \circledcirc$
3. $c * f \in \circledcirc$
4. $f + g \in \circledcirc$
5. $f * g \in \circledcirc$

Proof.

1. (a) Because f is spiralling, we can find a number $N \geq 0$ such that for all $m \geq 0$ we have for all $n \geq N$ that $f(n) \geq m$. For the shifted function $\mathcal{S}^c(f)$ we can take the same number $N \geq 0$.
   (b) Here too, we can take the same values $N \geq 0$ and $p > 0$ that exists because f is spiralling.
2. The proof is analogous to the proof of property 3.
3. (a) It is well known that, if $\lim_{n\to\infty} f(n) = \infty$ then $\lim_{n\to\infty} f(n) * c = \infty$.
   (b) Let $m > 0$, then there exists $p > 0$, $N \geq 0$ such that for all $n \geq N$ we have $f(n+p) \equiv f(n) \bmod m$, but then for the same p and N, we have $c * f(n+p) \equiv c * f(n) \bmod m$.
4. The proof is analogous to the proof of property 5.
5. (a) It is trivial that, if $\lim_{n\to\infty} f(n) = \infty$ and $\lim_{n\to\infty} g(n) = \infty$, then $\lim_{n\to\infty} f(n) * g(n) = \infty$.
   (b) Let $m > 0$. Because f and g are spiralling, we can find $p_f, p_g > 0$ and $N_f, N_g \geq 0$ such that $n \geq N_f$ implies $f(n + p_f) \equiv f(n) \bmod m$ and for all $n \geq N_g$ we have $g(n + p_g) \equiv g(n) \bmod m$. Take $N := max(N_f, N_g)$ and $p := p_f * p_g$, then for all $n \geq N$:

$$(f * g)(n+p) = f(n+p) * g(n+p) =$$
$$f(n + p_f p_g) * g(n + p_f p_g) \equiv f(n)g(n) = (f * g)(n) \bmod m \qquad \square$$

Theorem 12.4. Let $f, g \in \circledcirc$ with g increasing, then $f \circ g \in \circledcirc$

Proof. The first property of the definition is a well-known result from analysis. We will focus on the second property:
To Prove: $\forall m > 0\ \exists N \geq 0, p > 0\ \forall n \geq N \qquad (f \circ g)(n+p) \equiv (f \circ g)(n) \bmod m$
Choose any $m > 0$. Because f and g are spiralling, we can find $N_f, N_g \geq 0$ and $p_f, p_g > 0$ such that:

$$\forall n \geq N_f \qquad f(n + p_f) \equiv f(n) \bmod m \tag{12.5}$$
$$\forall n \geq N_g \qquad g(n + p_g) \equiv g(n) \bmod p_f \tag{12.6}$$

Choose $N \geq N_g$ such that $\forall n \geq N[g(n) > N_f]$. This is possible because $\lim_{n\to\infty} g(n) = \infty$. By equation (12.5) we have that for all $n \geq N$ that $f(n + k * p_f) \equiv f(n) \bmod m$.
By equation (12.6) we have that for all $n \geq N_g$ that $g(n + p_g) \equiv g(n) + k * p_f$ for some $k \geq 0$. If we combine this, we get the required result for $n \geq \max(N, N_g)$:

$$(f \circ g)(n + p_g) = f(g(n + p_g)) = f(g(n) + k * p_f) \equiv f(g(n)) = (f \circ g)(n) \bmod m$$

□

By application of Proposition 12.2 and Theorems 12.3 and 12.4, we find that the following functions are spiralling.

Corollary 12.7. More spiralling functions.

1. $P_k \in \circledcirc$ where P_k a polynomial of order k and positive coeficients.
2. $(n \mapsto 2^{2^n}), (n \mapsto 3^{3^{3^n}}) \in \circledcirc$

In particular, we know that $n \mapsto n^2$ is spiralling.

Example 12.8. Non-spiralling functions.
Finding a non-spiralling function that fails on the first requirement is easy. Take, for example: $n \mapsto 37$ or $n \mapsto \lfloor sin(n) \rfloor$. Finding a non-spiralling function that fails on the second requirement is a bit harder. For this, we must find a function f such that $\lim_{n\to\infty} f(n) = \infty$ but that is not periodic modulo some $m > 0$. We can take for example:

$$f(n) = \begin{cases} 2n + 1 & \text{if } n = k^2 \text{ for some } k > 0 \\ 2n & \text{otherwise} \end{cases}$$

If we consider $g := n \mapsto f(n) \bmod 2$, we find:

$$g(n) = \begin{cases} 1 & \text{if } n = k^2 \text{ for some } k > 0 \\ 0 & \text{otherwise} \end{cases}$$

We can easily see that g is not ultimately periodic modulo 2.

Lemma 12.9. Let $k > 0$ and $f_0, f_1, ..., f_{k-1} \in \circledcirc$. Then $zip_k(f_0, f_1, ..., f_{k-1}) \in \circledcirc$.

Proof. We have to show the two properties of Definition 12.1.

1. Choose $m \geq 0$. Because for all $i < k$ we have $f_i \in \circledcirc$, we can find $N_i \geq 0$ such that for all $n \geq N_i$ we have $f_i(n) \geq m$. Take $N := k * max(N_0, N_1, ..., N_{k-1})$. Then for all $n \geq N$ we can write $n = ky + i$ for $i < k$ and $y \geq 0$. We derive $zip_k(f_0, f_1, ..., f_{k-1})(n) = zip_k(f_0, f_1, ..., f_{k-1})(ky + i) = f_i(y) \geq m$.
2. Choose $m > 0$. Because for all $i < k$ we have $f_i \in \circledcirc$, we can find $N_i \geq 0$, $p_i > 0$ such that for all $n \geq N_i$ we have $f_i(n + p_i) \equiv f_i(n) \bmod m$. Let $N := k * max(N_0, N_1, ..., N_{k-1})$ and $\widetilde{p} := lcm(p_0, p_1, ..., p_{k-1})$, the least common multiple. Let $kn + i \geq N$, then:

$$zip_k(f_0, f_1, ..., f_{k-1})(kn + i) = f_i(n) \equiv$$
$$f_i(n + \widetilde{p}) = zip_k(f_0, f_1, ..., f_{k-1})(k(n + \widetilde{p}) + i) =$$
$$zip_k(f_0, f_1, ..., f_{k-1})(kn + i + \widetilde{p}k) \bmod m$$

   Thus zip_k is periodic modulo m with period $\widetilde{p}k$.

25

□

13 FST pumping lemma

A FST has only finitely many states, say $|Q|$. So whenever the FST has read more than $|Q|$ symbols, it must be in a state it has already been in. In this way, we can identify loops within the blocks of function streams.

Definition 13.1 ([1])**.** Zero loops.
Let $T := (Q, q_s, \delta, \lambda)$ be a FST.

1. A *zero loop* Z of a FST is a sequence of states $Z := (q_0, q_1, ..., q_{l-1})$ for some $l > 0$, such that any two subsequent states q_i and q_{i+1} are connected by a zero-transition. Furthermore q_{l-1} and q_0 are connected by a zero-transition too, so that the states form a loop when you only read zeros. Finally, no state may occur twice in Z. More formally:

$$\delta(q_j, \mathtt{0}) = q_{j+1} \qquad \text{(for all } j < l-1) \tag{13.2}$$
$$\delta(q_{l-1}, \mathtt{0}) = q_0 \tag{13.3}$$
$$\text{if } i \neq j \text{ then } q_i \neq q_j \qquad \text{(for all } i, j < l) \tag{13.4}$$

2. The *length* of a zero loop Z is the number $l > 0$ of distinct states in that loop.

3. Let $Z(T)$ be the least common multiple of the lengths of all zero loops of T. Notice that, because $|Q|$ is finite, we have finitely many zero loops. Furthermore, for every zero loop Z, we have that $\mathit{length}(Z) * k = Z(T)$ for some $k > 0$.

Before we prove some simple facts about zero loops, it is helpful to get a sense of what a zero loop is by looking at the illustration in Figure 13.5.

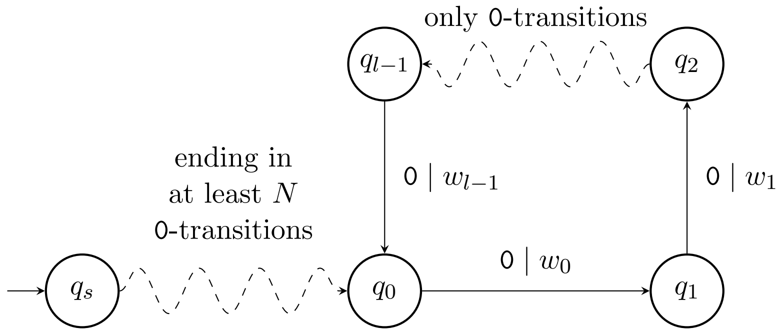


Figure 13.5: Illustration of a zero loop. The states $(q_0, ..., q_{l-1})$ form a zero loop.

We first notice that, after reading more than $|Q|$ zeros, any FST will be in a zero loop.

Proposition 13.6. Let $T := (Q, q_s, \delta, \lambda)$ be a FST. Then for all $w \in \mathbf{2}^*$, $q \in Q$ and whenever $n > |Q|$ we have $\delta(q, w \cdot \mathtt{0}^n) \in Z$ for some zero loop Z.

Proof. Without loss of generality, assume $w = \epsilon$ because we can replace q with $q_w := \delta(q, w)$. By the pigeonhole principle, and because $n > |Q|$, we have that some state, say $q_r \in Q$, has been visited twice. Thus $q_r = \delta(q, 0^r) = \delta(q, 0^{r+l})$ for some $r, l \geq 0$ with $r + l < n$. Given r, we can choose l to be the minimal number for which there is a state repetition. Then $Z = (q_r, \delta(q_r, 0), \delta(q_r, 0^2), ..., \delta(q_r, 0^{l-1}))$ is a zero loop, and $\delta(q, 0^n) \in Z$. $\square$

Next, we show some simple properties of zero loops. One can visualise them using Figure 13.5.

Proposition 13.7. Properties of zero loops.
Let $T := (Q, q_s, \delta, \lambda)$ be a FST. Let $Z := (q_0, q_1, ..., q_{l-1})$ be a zero loop with length $l > 0$. Let $t \geq 0$ and $w \in \mathbf{2}^*$.

1. For any $q_i \in Z$ we have $\delta(q_i, 0^{l*t}) = q_i$.
2. For any $q_i \in Z$ we have $\lambda(q_i, 0^{l*t}) = (\lambda(q_i, 0^l))^t$.
3. For any $q \in Q$ we have $\delta(q, w \cdot 0^{n+Z(T)*t}) = \delta(q, w \cdot 0^n)$ when $n > |Q|$.
4. For any $q \in Q$ we have $\lambda(q, w \cdot 0^{n+Z(T)*t}) = \lambda(q, w \cdot 0^n) \cdot (\lambda(q_r, 0^{l'}))^{t*k}$ for some $q_r \in Q$ and some $k, l' > 0$, when $n > |Q|$.

Proof.

1. Let $q_i \in Z$. Then we know that $\delta(q_i, 0) = q_{i+1}$ when $i < l - 1$ and $\delta(q_{l-1}, 0) = q_0$. So after reading l zeros, we are back in the same state: $\delta(q_i, 0^l) = q_i$. It is obvious that repeating this t times yields the result. We will, however, show that this can be proven more formal by means of induction on t. If $t = 0$, then $\delta(q_i, 0^{l*0}) = \delta(q_i, \epsilon) = q_i$. If we assume that $\delta(q_i, 0^{l*(t-1)}) = q_i$ (IH). Then we see that $\delta(q_i, 0^{l*t}) = \delta(q_i, 0^{l*(t-1)+l}) = \delta(\delta(q_i, 0^{l*(t-1)}), 0^l) \stackrel{IH}{=} \delta(q_i, 0^l) = q_i$.

2. Let $q_i \in Z$. We can use item 1 and see that:

$$\lambda(q_i, 0^{l*t}) = \lambda(q_i, 0^l) \cdot \lambda(\delta(q_i, 0^l), 0^l) \cdot ... \cdot \lambda(\delta(q_i, 0^{l*(t-1)}), 0^l) =$$
$$\underbrace{\lambda(q_i, 0^l) \cdot \lambda(q_i, 0^l) \cdot ... \cdot \lambda(q_i, 0^l)}_{t \text{ times}} = (\lambda(q_i, 0^l))^t$$

3. Let $q \in Q$. Then by Proposition 13.6 we know that $q_i := \delta(q, w \cdot 0^n) \in Z$ for some zero loop Z with length l. By Definition 13.1 we know $Z(T) = l * k$ for some $k > 0$ and by item 1 we have $\delta(q_i, 0^{l*(k*t)}) = q_i$. Thus $\delta(q, w \cdot 0^{n+Z(T)*t}) = \delta(q, w \cdot 0^{n+l*(k*t)}) = \delta(\delta(q, w \cdot 0^n), 0^{l*(k*t)}) = \delta(q, w \cdot 0^n)$.

4. Let $q \in Q$. By Proposition 13.6 we know that $q_r := \delta(q, w \cdot 0^n) \in Z$ for some zero loop Z with length l'. We know that $Z(T) = k * l'$ for some $k > 0$. Then by item 2 we have:

$$\lambda(q, w \cdot 0^{n+Z(T)*t}) = \lambda(q, w \cdot 0^n) \cdot \lambda(\delta(q, w \cdot 0^n), 0^{Z(T)*t}) =$$
$$\lambda(q, w \cdot 0^n) \cdot \lambda(q_r, 0^{l'*(t*k)}) = \lambda(q, w \cdot 0^n) \cdot (\lambda(q_r, 0^{l'}))^{t*k}$$

$\square$

After reading $|Q|$ zeros, a FST must be in a state repetition, so that we have entered a zero loop. Call the first repeated state q_r. If we continue reading more zeros, we remain in this loop. If we read $Z(T)$ more zeros, we are once again in state q_r. This idea resembles the pumping lemma for regular languages [11].

Lemma 13.8 ([8])**.** FST pumping lemma.
Let $T = (Q, q_s, \delta, \lambda)$ be a FST. For all $t \geq 0,\ q \in Q,\ n \geq |Q|$ we have $\delta(q, \mathtt{1}\mathtt{0}^{n+t*Z(T)}) = \delta(q, \mathtt{1}\mathtt{0}^n)$ and $\lambda(q, \mathtt{1}\mathtt{0}^{n+t*Z(T)}) = xy^t$ for some words $x, y \in \mathbf{2}^*$.

Proof. Because $n > |Q|$ we have by Proposition 13.7.3 that $\delta(q, \mathtt{1}\mathtt{0}^{n+t*Z(T)}) = \delta(q, \mathtt{1}\mathtt{0}^n)$
Furthermore, by Proposition 13.7.4 we have that:

$$\lambda(q, \mathtt{1}\mathtt{0}^{n+t*Z(T)}) = \lambda(q, \mathtt{1}\mathtt{0}^n) \cdot (\lambda(q_i, \mathtt{0}^l))^{kt} = \lambda(q, \mathtt{1}\mathtt{0}^n) \cdot ((\lambda(q_i, \mathtt{0}^l)^k))^t = xy^t$$

where $x := \lambda(q, \mathtt{1}\mathtt{0}^n)$ and $y := (\lambda(q_i, \mathtt{0}^l))^k$ for some $k, l > 0$ and $q_r \in Q$. $\square$

14 Transducts of spiralling function streams.

Definition 14.1. Block loop.
Let $T := (Q, q_s, \delta, \lambda)$ be a FST.
A *block loop* B of a function f and a FST is a sequence of states $B := (q_0, q_1, ..., q_{l-1})$ for $l > 0$, such that for some $N \geq 0$ and any $m \geq 0$:

$$\delta(q_i, \langle f(N + i + l * m)\rangle) = q_{i+1} \qquad \text{(for all } i < l - 1) \tag{14.2}$$

$$\delta(q_{l-1}, \langle f(N + l - 1 + l * m)\rangle) = q_0 \tag{14.3}$$

We call l the *length* of the block loop and N the *start index* of the block loop.

One can visualize block loops like zero loops, except that the transitions between states of the loop are caused by reading whole blocks of the function stream $\langle f\rangle$ instead of single symbols. We also allow state repetitions in block loops. An important thing to note is that the existence of block loops depends not only on the FST but also on the function f. Notice that the factor $m \geq 0$ is necessary. Without this, one cannot guarantee that the transitions between states of the block loop remain the same in the second and higher iterations of the loop! See Figure 14.4 for an illustration of a block loop.

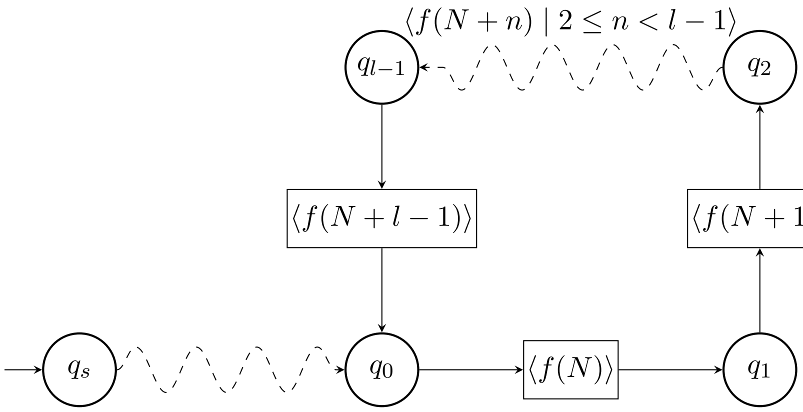


Figure 14.4: Illustration of the first iteration of a block loop. The states $(q_0, ..., q_{l-1})$ form a block loop.

Lemma 14.5. Let $T = (Q, q_s, \delta, \lambda)$ be a FST. Let $f \in$ 🌀. Then there exists a block loop $B = (q_0, q_1, ..., q_{l-1})$ of length l with start index N. Moreover, $q_0 = \delta(q_s, \langle f(n) \mid 0 \leq n < N\rangle)$ and for all $i \geq 0$, $n \geq N$ we have $f(n + li) \equiv f(n) \bmod Z(T)$.

Proof. Because f is spiralling, we know that there are N_{limit}, $N_{\text{periodic}} \geq 0$, $p > 0$ such that:

$$f(n) \geq |Q| \qquad \text{(for all } n \geq N_{\text{limit}}) \tag{14.6}$$

$$f(n + p) \equiv f(n) \bmod Z(T) \qquad \text{(for all } n \geq N_{\text{periodic}}) \tag{14.7}$$

Let $s_m \in Q$ be shorthand for $s_m := \delta(q_s, \langle f(n) \mid 0 \leq n < m\rangle)$. This is the state that T is in before reading the m-th occurrence of a one: the start of the m-th block.

By the pigeonhole principle, we can find $N \geq \max(N_{\text{limit}},\ N_{\text{periodic}}),\ b > 0$ such that $s_{N+p*b} = s_N$.
We claim that $B = (s_N, s_{N+1}, ..., s_{N+pb-1})$ is a block loop of length b.
Notice that by equation (14.7) we have that for any $m \geq 0$ and for some $k \geq 0$:

$$f(n + pb * m) = f(n) + k * Z(T) \qquad \forall n \geq N \tag{14.8}$$

We will show equation (14.2). Let $m \geq 0$ then, by Lemma 13.8 we have for all $b \geq 0$ and $s_{N+i} \in B$ that:

$$\begin{gathered}
\delta(s_{N+i}, \langle f(N + i + pb * m)\rangle) = \delta(s_{N+i}, \mathtt{1}\mathtt{0}^{f(N+i+pb*m)}) \overset{14.8}{=} \\
\delta(s_{N+i}, \mathtt{1}\mathtt{0}^{f(N+i)+k*Z(T)}) \overset{13.8}{=} \delta(s_{N+i}, \mathtt{1}\mathtt{0}^{f(N+i)}) = \\
\delta(s_{N+i}, \langle f(N + i)\rangle) = s_{N+i+1}
\end{gathered} \tag{14.9}$$

The last equality follows from the definition of s_m. We can thus see that, when reading blocks of a spiralling function, the transitions between the blocks remain the same from some point on, due to Lemma 13.8. Equation (14.3) follows directly from equation (14.9) and the fact that $s_{N+p*b} = s_N$.

$$\delta(s_{N+pb-1}, \langle f(N + pb - 1 + pb * m)\rangle) \overset{14.9}{=} s_{N+pb} = s_N$$

$\square$

Theorem 14.10 ([8])**.** Transducts of spiralling function streams.
Let $f \in \circledcirc$. Then $\langle f\rangle \geq \sigma$ if and only if

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)} \qquad \text{where} \qquad \varphi(i,j) := \frac{f(N + li + j) - a_j}{z} \in \mathbb{N}$$

where we have $N \geq 0,\ z, l > 0$, a list of numbers $(a_j)_{j=0}^{l-1} \in \mathbb{N}^l$, a word $w \in \mathbf{2}^*$ and lists of words $(x_j)_{j=0}^{l-1}, (y_j)_{j=0}^{l-1} \in (\mathbf{2}^*)^l$.

Proof. ($\Longrightarrow$) If $\langle f\rangle \geq \sigma$ then $\langle f\rangle \geq_T \sigma$, by some transducer $T := (Q, q_s, \delta, \lambda)$. By Lemma 14.5 we know that there exists a block loop $B := (q_0, q_1, ..., q_{l-1})$ with start index N and length l such that:

$$q_0 = \delta(q_s, \langle f(n) \mid 0 \leq n < N\rangle) \tag{14.11}$$

$$f(n + li) \equiv f(n) \bmod Z(T) \qquad \text{(for all } i \geq 0,\ n \geq N) \tag{14.12}$$

We first rewrite σ. Note that we only apply the definition of a function stream and the definition of the output funciton λ. At $(*)$ we use the definition of a block loop.

$$\sigma = \prod_{i=0}^{\infty} \lambda(\delta(q_s, \langle f(n) \mid 0 \leq n < i \rangle), \mathtt{1}\mathtt{0}^{f(i)}) =$$

$$\lambda(q_s, \langle f(n) \mid 0 \leq n < N \rangle) \cdot \prod_{i=0}^{\infty} \lambda(q_0, \langle f(N + il + n) \mid 0 \leq n < l \rangle) =$$

$$w \cdot \prod_{i=0}^{\infty} \lambda(q_0, \prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{f(N+il+j)}) \overset{(*)}{=} \tag{14.13}$$

$$w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} \lambda(q_j, \mathtt{1}\mathtt{0}^{f(N+il+j)})$$

$$\text{where } w := \lambda(q_s, \langle f(n) \mid 0 \leq n < N \rangle)$$

Define $a_j := \min\{f(N + il + j) \mid i \geq 0\}$. Then:

$$f(N + il + j) \geq a_j \qquad \text{(for all } i \geq 0 \text{ and } j < l) \tag{14.14}$$

Let $z := Z(T)$. We write $f(N + il + j)$ into another form by using equation (14.12):

$$\begin{aligned}
f(N + il + j) &\equiv a_j \bmod Z(T) && \Longrightarrow \\
f(N + il + j) &= a_j + k_{i,j} * Z(T) \text{ for some } k_{i,j} \in \mathbb{N} && \Longrightarrow \\
k_{i,j} * Z(T) &= f(N + il + j) - a_j && \Longrightarrow \\
k_{i,j} &= \frac{f(N + il + j) - a_j}{Z(T)} && \Longrightarrow \\
k_{i,j} &= \frac{f(N + il + j) - a_j}{z} =: \varphi(i, j)
\end{aligned}$$

Notice that $k_{i,j} \in \mathbb{N}$ instead of $k_{i,j} \in \mathbb{Z}$ because of equation (14.14).
Then we finally have by the FST pumping lemma (13.8) that:

$$\sigma \overset{14.13}{=} w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} \lambda(q_j, \mathtt{1}\mathtt{0}^{f(N+il+j)}) = w \cdot \prod_{i=0}^{\infty} \prod_{m}^{l-1} \lambda(q_j, \mathtt{1}\mathtt{0}^{a_j + k_{i,j} * Z(T)}) =$$

$$w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} \lambda(q_j, \mathtt{1}\mathtt{0}^{a_j + \varphi(i,j) * Z(T)}) \overset{13.8}{=} w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j y_j^{\varphi(i,j)}$$

$(\Longleftarrow)$ Let $\sigma := w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)}$ where $\varphi(i, j)$ is of the form described above, so we have a list of numbers $(a_j)_{j=0}^{l-1} \in \mathbb{Z}^l$, a number $N \geq 0$ and a number $z > 0$. Futhermore, $(x_j)_{j=0}^{l-1}, (y_j)_{j=0}^{l-1} \in (\mathbf{2}^*)^l$ are lists of words and $w \in \mathbf{2}^*$ is a word. We want to show that $\langle f \rangle \geq w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{m} x_j \cdot y_j^{\varphi(i,j)}$. We do this in several steps:

1. $\langle f \rangle \geq \langle n \mapsto f(N + n) \rangle$
   by Lemma 9.5.1

2. $\langle n \mapsto f(N+n)\rangle \geq \langle n \mapsto f(N+n) - a_{n \bmod l}\rangle$
Let $j := n \bmod l$. We want to show that we can find a transducer that removes a_j zeros from the j-th block. Define $T := (Q, q^0_{l-1}, \delta, \lambda)$. Let $Q := \{q_{j,h} \mid j < l, h \leq a_j\}$. Define for all $j < l$, the transition and output functions as follows:

$$\begin{aligned}
\delta(q_{j,0}, \mathtt{0}) &:= q_{j,0} & \lambda(q_{j,0}, \mathtt{0}) &:= \mathtt{0} \\
\delta(q_{j,0}, \mathtt{1}) &:= q_{(j+1 \bmod l),a_j} & \lambda(q_{j,0}, \mathtt{1}) &:= \mathtt{1} \\
\delta(q_{j,h}, \mathtt{0}) &:= q_{j,h-1} & \lambda(q_{j,0}, \mathtt{0}) &:= \epsilon \qquad \text{(for all } 0 < h \leq a_j)
\end{aligned}$$

Note that this partial description is sufficient, because we assume that $\langle n \mapsto f(N+n)\rangle$ is given as input. See Figure 14.15 for an illustration.

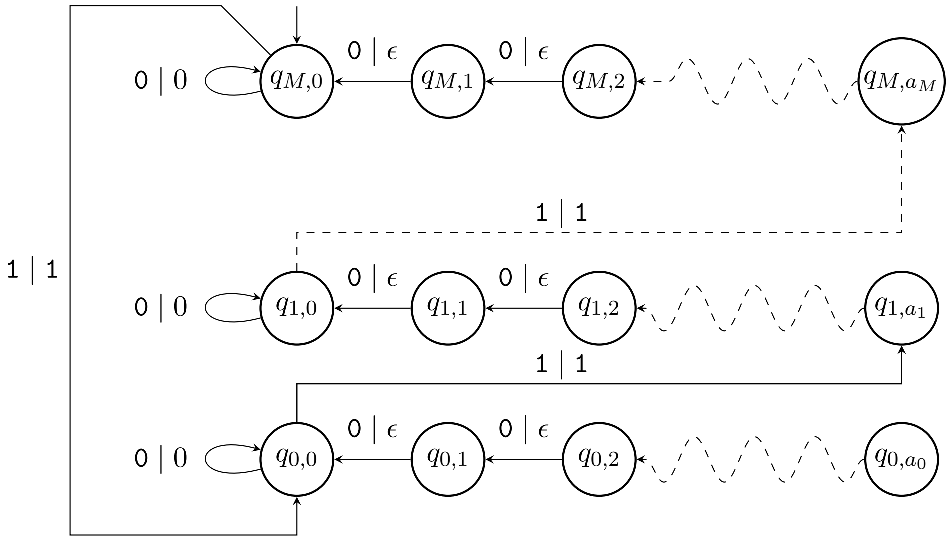


Figure 14.15: The FST T described above where $M := l-1$.

3. $\langle n \mapsto f(N+n) - a_{n \bmod l}\rangle \geq \prod_{i=0}^{\infty}\prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\varphi(i,j)}$ We give a derivation below. Note that we only introduce a division by z. This is possible by Lemma 9.5.2.

$$\langle n \mapsto f(N+n) - a_{n \bmod l}\rangle = \prod_{n=0}^{\infty} \mathtt{1}\mathtt{0}^{f(N+n)-a_{n \bmod l}} =$$
$$\prod_{i=0}^{\infty}\prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{f(N+i+j)-a_j} \overset{9.5.2}{\geq} \prod_{i=0}^{\infty}\prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\frac{f(N+i+j)-a_j}{z}} =$$
$$\prod_{i=0}^{\infty}\prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\varphi(i,j)}$$

4. $\prod_{i=0}^{\infty}\prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\varphi(i,j)} \geq \prod_{i=0}^{\infty}\prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)}$. We must periodically replace $\mathtt{1}$ by x_j and $\mathtt{0}$ by y_j. We give a FST to do this in the proof of Theorem 15.9.

5. $\prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)} \geq w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j \cdot y_j^{\varphi(i,j)}$.
   By Proposition 8.1.4.

$\square$

15 Transition ambiguities

By Theorem 14.10 we have seen that we can write the transducts of a spiralling function stream as a double product:

$$\langle f \rangle \geq w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j y_j^{\varphi(i,j)}$$

for lists of words $(x_j)_{j=0}^{l-1}, (y_j)_{j=0}^{l-1} \in (\mathbf{2}^*)^l$ a number $l > 0$ a function φ and a word $w \in \mathbf{2}^*$. We want to simplify this representation without changing the degree of the transduct. It would be nice if we could replace x_j by 1 and y_j by 0. In this section we will show that this is indeed possible by showing the following equivalence:

$$w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j y_j^{\varphi(i,j)} \equiv \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} \mathtt{1}\mathtt{0}^{\varphi'(i,j)}$$

for a different function φ' of the same form as φ and number $l' > 0$.
It is easy to construct a FST that transduces the right stream into the left stream ($\leq$). The other direction seems just as simple, but there is a difficulty here. It is possible that there are *transition ambiguities*: points where a FST cannot detect the transition form $y_j^{\varphi(i,j)}$ to $x_{j+1} y_{j+1}^{\varphi(i,j+1)}$. We will first have to resolve these ambiguities, before we can show this equivalence.

Definition 15.1. Transition ambiguities.
Let $\langle f \rangle \geq \sigma$ with $f \in 🌀$ and σ written as in Theorem 14.10:

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} x_j y_j^{\varphi(i,j)}$$

for a number $l > 0$, lists of words $(x_j)_{j=0}^{l-1}, (y_j)_{j=0}^{l-1} \in (\mathbf{2}^*)^l$ a function φ of the form described in Theorem 14.10 and a word $w \in \mathbf{2}^*$. Then we say that this product *has transition ambiguities* if one of the following holds:

$$\exists j < l-1 \qquad y_j^{\omega} = x_{j+1} y_{j+1}^{\omega} \tag{15.2}$$

$$y_{l-1}^{\omega} = x_0 y_0^{\omega} \tag{15.3}$$

$$\exists j < l \qquad x_j = \epsilon \tag{15.4}$$

Example 15.5. A transition ambiguity.
Consider the words $u, x, y := \mathtt{10}$. Then $u^{\omega} = (\mathtt{10})^{\omega} = \mathtt{10}(\mathtt{10})^{\omega} = xy^{\omega}$. A FST cannot recognise the transition between u and xy because they look "the same" as streams. To illustrate this problem even more, consider the following stream with $x_0, x_1, y_0, y_1 := \mathtt{10}$:

$$\sigma := \prod_{i=0}^{\infty} \prod_{j=0}^{1} x_i y_i^{\varphi(i,j)} = \prod_{i=0}^{\infty} x_0 y_0^{\varphi(i,0)} x_1 y_1^{\varphi(i,1)}$$

A FST cannot detect if $\varphi(i,0) = 2$ and $\varphi(i,1) = 3$ or $\varphi(i,0) = 3$ and $\varphi(i,1) = 2$ for some $i \geq 0$.

$$x_0 y_0^2 x_1 y_1^3 = \mathtt{10}(\mathtt{10})^2\mathtt{10}(\mathtt{10})^3 = (\mathtt{10})^7 = \mathtt{10}(\mathtt{10})^3\mathtt{10}(\mathtt{10})^2 = x_0 y_0^3 x_1 y_1^2$$

We will resolve ambiguities of the form given in equations (15.2) and (15.3) in the next lemma.

Lemma 15.6. Let $u, x, y \in \mathbf{2}^*$ with $u, y \neq \epsilon$ and $i, j \geq 0$ such that $u^i = xy^j$. Then there exists a word $v \in \mathbf{2}^*$ and $a, b \geq 0$ such that $u^m x y^n = x v^{am+bn}$ for all $m, n \geq 0$.

Proof. We refer to [8]. □

Example 15.7. We use Lemma 15.6 to resolve the transition ambiguity posed in Example 15.5. Let $u, x, y := \mathtt{10}$, then $u^2 = xy^1$. We can choose $v := \mathtt{10}$ and $a, b := 1$. Then we have for all $m, n \geq 0$ that $(\mathtt{10})^m(\mathtt{10})(\mathtt{10})^n = (\mathtt{10})(\mathtt{10})^{m+n}$. The essential difference is that we have merged two blocks together without changing the resulting stream. To clarify, we have merged two blocks $x_j y_j^{\varphi(i,j)}$ and $x_{j+1} y_{j+1}^{\varphi(i,j+1)}$ together into one new block $\widetilde{x_j}\widetilde{y_j}^{\varphi'(i,j)}$. Note that we also need to change the function φ to a function φ' in order for this to work. That this can be done without changing the stream is shown in the Lemma 15.8.

Lemma 15.8. Let $f \in \circlearrowleft$ and $\langle f \rangle \geq \sigma$. If we write σ in the form of Theorem 14.10, then we can find a number $l' > 0$, lists of words $(\widetilde{x_j})_{j=0}^{l'-1}, (\widetilde{y_j})_{j=0}^{l'-1} \in (\mathbf{2}^*)^l$, a word $\widetilde{w} \in \mathbf{2}^*$ and φ' such that

$$\sigma = \widetilde{w} \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l'-1} \widetilde{x_j}\widetilde{y_j}^{\varphi'(i,j)}$$

contains no transition ambiguities (Definition 15.1) and φ' is of the form described in Theorem 14.10.

Proof. For this we refer to [8]. The idea is to iteratively remove transition ambiguities of the form given in equations (15.2) and (15.3) by means of Lemma 15.6. In a similar manner ambiguities of the form given in equation (15.4) can be removed. □

Theorem 15.9. Let $f \in \circlearrowleft$. Then $\langle f \rangle \geq \sigma$ if and only if

$$\sigma \equiv \prod_{i=0}^{\infty} \prod_{j=0}^{l'-1} \mathtt{10}^{\varphi'(i,j)} \qquad \text{where} \qquad \varphi'(i,j) = \frac{f(N + il' + j) - \widetilde{a_j}}{z} \in \mathbb{N}$$

where $\widetilde{a_j} \geq 0$ for $j < l$ and $z, l' > 0,\ N \geq 0$.

Proof. If $f \in \circlearrowleft$ and $\langle f \rangle \geq \sigma$, then by Theorem 14.10 and Lemma 15.8 we can write

$$\sigma = \widetilde{w} \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l'-1} \widetilde{x_j}\widetilde{y_j}^{\varphi'(i,j)} \qquad \text{where} \qquad \varphi'(i,j) := \frac{f(N + l'i + j) - a_j}{z} \in \mathbb{N} \tag{15.10}$$

for some integer $N \geq 0$, integers $z, l' > 0$, a list of numbers $(\widetilde{a_j})_{j=0}^{l'-1} \in (\mathbb{N})^{l'-1}$, a word $\widetilde{w} \in \mathbf{2}^*$ and lists of words $(\widetilde{x_j})_{j=0}^{l'-1}, (\widetilde{y_j})_{j=0}^{l'-1} \in (\mathbf{2}^*)^{l'}$, such that equation (15.10) contains no transition ambiguities. We need to show that $\sigma \geq \prod_{i=0}^{\infty} \prod_{j=0}^{l'-1} \mathtt{10}^{\varphi'(i,j)}$ and that $\sigma \leq \prod_{i=0}^{\infty} \prod_{j=0}^{l'-1} \mathtt{10}^{\varphi'(i,j)}$

To Prove: $(\geq)$

For this we refer to [8]. The idea is to find a kind of *markers* to notice the transition between blocks of the stream. It is possible to find these markers because there are no transition ambiguities.

To Prove: $(\leq)$

We create a FST T that applies a periodic encoding to σ (see Figure 15.11). Prepending $\widetilde{w}$ is possible by Proposition 8.1.4. Define $T := (Q, q_{l'-1}, \delta, \lambda)$ where $Q := \{q_0, q_1, ..., q_{l'-1}\}$ and δ and λ defined for $i < l'$ by:

$$\delta(q_i, 0) := q_i \qquad \lambda(q_i, 0) := \widetilde{y}_i$$
$$\delta(q_i, 1) := q_{i+1 \bmod l'} \qquad \lambda(q_i, 1) := \widetilde{x}_{i+1 \bmod l'}$$

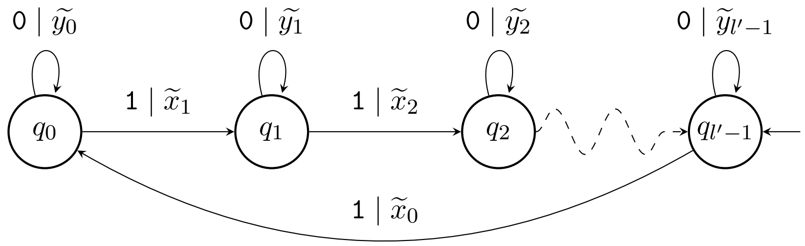


Figure 15.11: Transducer that applies a periodic encoding to a stream.

Together we get the required result:

$$\prod_{i=0}^{\infty}\prod_{j=0}^{l'-1} 10^{\varphi'(i,j)} \geq_T \prod_{i=0}^{\infty}\prod_{j=0}^{l'-1} \widetilde{x}_j \widetilde{y}_j^{\varphi'(i,j)} \overset{8.1.4}{\geq} \widetilde{w} \cdot \prod_{i=0}^{\infty}\prod_{j=0}^{l'-1} \widetilde{x}_j \widetilde{y}_j^{\varphi'(i,j)}$$

$\square$

16 Masses

We introduce *masses* to simplify the function $\varphi(i, j)$ in Theorems 14.10 and 15.9. We shorten the double product of these theorems to a much more elegant form using *mass products*. We start by giving a number of definitions.

Definition 16.1 ([9])**.** Weights and masses.

1. We define a *weight* as a tuple $\vec{\alpha} := [a_0, ..., a_{n-1}] \in \mathbb{Q}^n$.
2. We define the *length* of a weight as $|\vec{\alpha}| := n$.
3. We say that a weight $\vec{\alpha}$ is *positive* if $\vec{\alpha}_i \geq 0$ for all $i < |\vec{\alpha}|$.
4. We say that a weight $\vec{\alpha}$ is *constant* if $\vec{\alpha}_i = 0$ for all $i < |\vec{\alpha}|$.
5. We define a *mass* as a tuple of positive weights $\vec{\boldsymbol{\alpha}} = (\vec{\alpha}_0, ..., \vec{\alpha}_{l-1})$.
6. We define the *length* of a mass as $|\vec{\boldsymbol{\alpha}}| := l$.
7. We say that a mass $\vec{\boldsymbol{\alpha}}$ is *constant* if $\vec{\boldsymbol{\alpha}}_i$ is constant for all $i < |\vec{\alpha}|$.
8. We define $flat(\vec{\boldsymbol{\alpha}})$ as the concatination of all weights of $\vec{\boldsymbol{\alpha}}$. For example $flat(([1, 2], [3, 4])) = [1, 2, 3, 4]$.
9. We define the flattened length $\|\vec{\boldsymbol{\alpha}}\|$ as the sum of the lengths of all weights in a mass:
$$\|\vec{\boldsymbol{\alpha}}\| := \sum_{i=0}^{|\vec{\boldsymbol{\alpha}}|} |\vec{\boldsymbol{\alpha}}_i|$$

Definition 16.2 ([9])**.** Weight and mass operations.
Let $f : \mathbb{N} \to \mathbb{N}$ be a natural function and $n \geq 0$.

1. We define the rotation of a weight $\vec{\beta}$ with $|\vec{\beta}| = l$ recursively:
$$\vec{\beta}^{(0)} := \vec{\beta}$$
$$\vec{\beta}^{(n+1)} := [\vec{\beta}_{l-1}, \vec{\beta}_0, ..., \vec{\beta}_{l-2}]^{(n)}$$
2. We define the rotation of a mass $\vec{\boldsymbol{\alpha}}$ with $|\vec{\boldsymbol{\alpha}}| = l$ recursively:
$$\vec{\boldsymbol{\alpha}}^{(0)} := \vec{\boldsymbol{\alpha}}$$
$$\vec{\boldsymbol{\alpha}}^{(n+1)} := (\vec{\boldsymbol{\alpha}}_{l-1}, \vec{\boldsymbol{\alpha}}_0, ..., \vec{\boldsymbol{\alpha}}_{l-2})^{(n)}$$
3. We define the *scalar product* of a weight $\vec{\alpha} = [a_0, ..., a_{q-1}]$ for some $q > 0$ as:
$$\vec{\alpha} \odot f := \sum_{i=0}^{q-1} a_i * f(i) = a_0 * f(0) + a_0 * f(1) + ... + a_{q-1} * f(q-1)$$
4. We define the *weight displacement* of a weight $\vec{\beta} = [a_0, ..., a_{q-1}]$. Notice that we can write $n = qk + i$ for $i < q$ and $k \geq 0$.
$$(\vec{\beta} \oplus f) : \mathbb{N} \to \mathbb{Q}$$
$$(\vec{\beta} \oplus f)(n) := (\vec{\beta} \oplus f)(qk + i) = a_i + f(qk + i)$$

5. We define the *mass product* of a mass $\vec{\boldsymbol{\alpha}}$ recursively:

$$\begin{aligned}(\vec{\boldsymbol{\alpha}} \otimes f) &: \mathbb{N} \to \mathbb{Q}\\ (\vec{\boldsymbol{\alpha}} \otimes f)(0) &:= \vec{\boldsymbol{\alpha}}_0 \odot f\\ (\vec{\boldsymbol{\alpha}} \otimes f)(n+1) &:= (\vec{\boldsymbol{\alpha}}^{(1)} \otimes \mathcal{S}^{|\vec{\boldsymbol{\alpha}}_0|}(f))(n)\end{aligned}$$

6. We say that $(\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))$ is natural if $(\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))(n) \in \mathbb{N}$ for all $n \geq 0$ and likewise that $(\vec{\boldsymbol{\alpha}} \otimes f)$ is natural if $(\vec{\boldsymbol{\alpha}} \otimes f)(n) \in \mathbb{N}$ for all $n \geq 0$.

7. We say that a weight $\vec{\beta}$ is constant, if $\vec{\beta}_i = 0$ for all $i < |\vec{\beta}|$.

8. We say that a mass $\vec{\boldsymbol{\alpha}}$ is constant, if al its weights are constant.

Example 16.3. We will show $\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f)$ for the function $f(n) = 2n$, the mass $\vec{\boldsymbol{\alpha}} := ([0, 1, \frac{1}{2}], [3])$ and the weight $\vec{\beta} := [2, -8]$. Notice that $\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f)$ is natural, even though the mass contains a fraction and the weight contains a negative number.

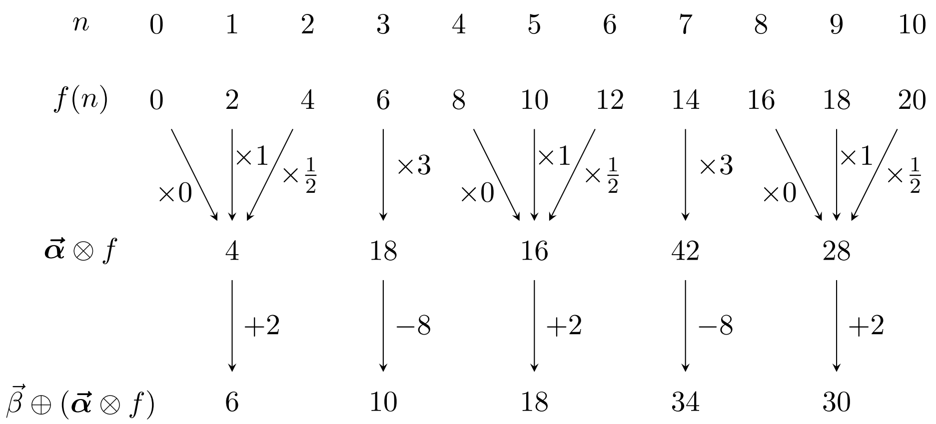


Figure 16.4: Example of a weight displacement applied to a mass product.

Lemma 16.5 ([8])**.** Let $f : \mathbb{N} \to \mathbb{N}$ be a function. Let $\vec{\boldsymbol{\alpha}}$ be a non-constant mass with $m := |\vec{\boldsymbol{\alpha}}|$. Then there exists a list of non-constant weights $(\vec{\beta}_i)_{i=0}^{m}$ such that:

$$\vec{\boldsymbol{\alpha}} \otimes f = zip_m(g_0, g_1, \ldots, g_{m-1}) \qquad \text{where} \qquad g_i = ((\vec{\beta}_i) \otimes f) \text{ for } i < m$$

Proof. We only show the basic idea. Let $\vec{\beta}$ be a mass such that $|\vec{\beta}_i| = \|\vec{\boldsymbol{\alpha}}\|$ for all $i < m$. Define $s_l := \sum_{i=0}^{l} |\vec{\boldsymbol{\alpha}}_i|$ for $l < m$. Define $\vec{\beta}$ by

$$\vec{\boldsymbol{\beta}}_{i, s_j + h} := \begin{cases} \vec{\boldsymbol{\alpha}}_{i,h} & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases} \qquad \text{where } j < m \text{ and } h < |\vec{\boldsymbol{\alpha}}_i|$$

For example, if $\vec{\boldsymbol{\alpha}} = ([1, 2, 3], [1, 2], [0, 1, 0])$ then

$$\vec{\boldsymbol{\beta}} = \begin{pmatrix} [1, 2, 3, 0, 0, 0, 0, 0], \\ [0, 0, 0, 1, 2, 0, 0, 0], \\ [0, 0, 0, 0, 0, 0, 1, 0] \end{pmatrix}$$

$\square$

Proposition 16.6 ([8])**.** Let $f \in \circledS$, let $\vec{\boldsymbol{\alpha}}$ be a non-constant mass and $\vec{\beta}$ a weight. Then we have:

1. $\vec{\beta} \oplus f \in \circledS$ if $\vec{\beta} \oplus f$ is natural.
2. If $\vec{\beta}$ is positive and $(\vec{\beta}) \otimes f$ is natural then $(\vec{\beta}) \otimes f \in \circledS$.
3. $\vec{\boldsymbol{\alpha}} \otimes f \in \circledS$ if $\vec{\boldsymbol{\alpha}} \otimes f$ is natural.
4. $\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f) \in \circledS$ if $\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f)$ is natural.

Proof.
Let $q := |\vec{\beta}|$ and $l := |\vec{\boldsymbol{\alpha}}|$ be the lengths of the weight and mass respectively.

1. We show properties (a) and (b) of Definition 12.1.

   (a) Let $m \geq 0$. Let $M := \min(\vec{\beta}_0, \vec{\beta}_1, \ \ldots, \vec{\beta}_{q-1})$. Because $f \in \circledS$, we have, for some $N \geq 0$, that $f(n) \geq |M| + m$ for all $n \geq N$. Write $n = qk + i$ for $i < q$ and $k \geq 0$.

   $$(\vec{\beta} \oplus f)(qk+i) = \vec{\beta}_i + f(qk+i) \geq \vec{\beta}_i + |M| + m \geq m$$

   (b) Let $m > 0$. Then because $f \in \circledS$ we have $N \geq 0$, $p > 0$ such that for all $n \geq N$ we have $f(n+p) \equiv f(n) \bmod m$. Let $\widetilde{p} := q * p$. Write $n = qk + i$ for $i < q$ and $k \geq 0$.

   $$(\vec{\beta} \oplus f)(n) = (\vec{\beta} \oplus f)(qk+i) = \vec{\beta}_i + f(qk+i) \equiv$$
   $$\vec{\beta}_i + f(qk+i+pq) = (\vec{\beta} \oplus f)(n + \widetilde{p}) \bmod m$$

2. We show properties (a) and (b) of Definition 12.1. We can say

   $$\vec{\beta} = [\frac{a_0}{d}, \frac{a_1}{d}, \ \ldots, \frac{a_{q-1}}{d}]$$

   where $a_j \geq 0$ for all $j < q$ (because $\vec{\beta}$ is positive) and $d > 0$. Because the mass "$(\vec{\beta})$" only contains a single weight "$\vec{\beta}$" we can easily give an explicit form of the mass product:

   $$((\vec{\beta}) \otimes f)(n) = \vec{\beta} \odot \mathcal{S}^{nq}(f) =$$
   $$\sum_{i=0}^{q-1} \vec{\beta}_i * (\mathcal{S}^{nq}(f))(i) = \sum_{i=0}^{q-1} \vec{\beta}_i * f(nq+i) = \tag{16.7}$$
   $$\sum_{i=0}^{q-1} \frac{a_i}{d} * f(nq+i)$$

   (a) Let $m \geq 0$, then by $f \in \circledS$, we can find $N \geq 0$ such that for all $n \geq N$ we have $f(n) \geq \frac{m}{b*q}$ where $b := \min(\frac{a_0}{d}, \frac{a_1}{d}, \ \ldots, \frac{a_{q-1}}{d})$. Because $\vec{\beta}$ is positive, we have for $n \geq N$,

   $$((\vec{\beta}) \otimes f)(n) \stackrel{16.7}{=} \sum_{i=0}^{q-1} \frac{a_i}{d} * f(nq+i) \geq$$
   $$\sum_{i=0}^{q-1} b * f(nq+i) \geq \sum_{i=0}^{q-1} b * \frac{m}{b*q} = m$$

(b) Let $m > 0$, then by $f \in \circledcirc$, we can find $N \geq 0,\ p > 0$ such that:

$$f(n+p) \equiv f(n) \bmod dm \tag{16.8}$$

Then we have for $(k_i) \in (\mathbb{Z})^q$ that:

$$((\vec{\beta}) \otimes f)(n) \overset{16.7}{=} \sum_{i=0}^{q-1} \frac{a_i}{d} * f(nq+i) \overset{16.8}{=}$$
$$\sum_{i=0}^{q-1} \frac{a_i}{d} * (f(nq+i+pq) + k_i md) =$$
$$\sum_{i=0}^{q-1} \frac{a_i}{d} * (f(nq+i+pq)) + \sum_{i=0}^{q-1} \frac{a_i}{d} * (k_i md) =$$
$$\sum_{i=0}^{q-1} \frac{a_i}{d} * (f(nq+i+pq)) + \sum_{i=0}^{q-1} a_i(k_i m) \equiv$$
$$\sum_{i=0}^{q-1} \frac{a_i}{d} * (f(q(n+p)+i)) \overset{16.7}{=} ((\vec{\beta}) \otimes f)(n+p) \bmod m$$

3. Let $l := |\vec{\boldsymbol{\alpha}}|$, then we have by Lemma 16.5 that:

$$\vec{\boldsymbol{\alpha}} \otimes f = zip_l(g_0, g_1, \ \ldots, g_{l-1}) \qquad \text{where} \qquad g_i = ((\vec{\boldsymbol{\beta}}_i) \otimes f) \text{ for } i < l$$

for positive weights $\vec{\boldsymbol{\beta}}_i$. In the previous item, we have seen that $g_i \in \circledcirc$ for all $i < l$. The result follows from the fact that *zip* is spiralling when all its arguments are spiralling as shown in Lemma 12.9.

4. This follows from the first and third item.

$\square$

Lemma 16.9. [8] If $f \in \circledcirc$ then $\langle f \rangle \geq \langle \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \rangle$ for any mass $\vec{\boldsymbol{\alpha}}$, weight $\vec{\beta}$ and $k \geq 0$ if $\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f))$ is natural.

Proof. We give a sketch of this proof. Let $f \in \circledcirc$. Because $(\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \in \circledcirc$ by Proposition 16.6, we can seperately show that $\langle f \rangle \geq \vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)$ and $\langle f \rangle \geq \beta \oplus f$. We can also ignore the shift of f because $\mathcal{S}^k(f) \in \circledcirc$ by Theorem 12.3 so that $\mathcal{S}^k(f) \equiv \langle f \rangle$ by Lemma 9.5.
To Prove: $\langle f \rangle \geq \vec{\boldsymbol{\alpha}} \otimes f$

We start by writing out the mass $\vec{\boldsymbol{\alpha}}$.

$$\vec{\boldsymbol{\alpha}} = (\vec{\alpha}_0, \vec{\alpha}_1, ..., \vec{\alpha}_{l-1}) \qquad\qquad \text{where}$$
$$\vec{\alpha}_i = [\frac{a_{i,0}}{d_i}, \frac{a_{i,1}}{d_i}, ..., \frac{a_{i,q_i}}{d_i}] \qquad \text{for } i < |\vec{\boldsymbol{\alpha}}| \text{ and where } q_i := |\vec{\alpha}_i|$$

We will create three seperate transducers T_1, T_2 and T_3.

1. T_1 multiplies each value of f by $a_{i,j}$ in a periodic way, such that:

$$\langle f \rangle \geq_{T_1} \langle n \mapsto (flat(\vec{\boldsymbol{\alpha}})_{n \bmod \|\vec{\boldsymbol{\alpha}}\|}) * f(n) \rangle$$

2. T_2 adds the values together that belong to one weight.

3. T_3 divides the values of the resulting function by d_i.

To Prove: $\langle f \rangle \geq \vec{\beta} \oplus f$
This follows from a similar construction as shown in Figure 14.15 combined with T_3 of the previous proof sketch.

$\square$

Theorem 16.10 ([8])**.** If $f \in$, then $\langle f \rangle \geq \sigma$ if and only if $\sigma \equiv \langle \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \rangle$ for some integer $k \geq 0$, a mass $\vec{\boldsymbol{\alpha}}$ and a weight $\vec{\beta}$.

Proof. ($\implies$) By Theorem 15.9 we have that

$$\sigma \equiv \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} 10^{\varphi(i,j)} \qquad \text{where} \qquad \varphi(i,j) := \frac{f(N+li+j) - a_j}{z}$$

for a number $l, z > 0$ and $a_j \geq 0$ for all $j < l$. Let $\vec{\beta} := [-\frac{a_0}{z}, -\frac{a_1}{z}, ..., -\frac{a_{j-1}}{z}]$ and $\vec{\boldsymbol{\alpha}} := ([z])$. Without loss of generality assume N to be a multiple of l. For all $n \geq N$ write $n = N + il + j$ where $j < l$ and $i \geq 0$.

$$\begin{gathered}(\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))(n) = (\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))(N+il+j) = \\ (\vec{\boldsymbol{\alpha}} \otimes f)(N+il+j) - \frac{a_j}{z} = \frac{1}{z} * f(N+il+j) - \frac{a_j}{z} = \\ \frac{f(N+il+j)}{z} - \frac{a_j}{z} = \frac{f(N+il+j) - a_j}{z} = \varphi(i,j)\end{gathered}$$

We can use this derivation to write:

$$\begin{gathered}\sigma \equiv \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} 10^{\varphi(i,j)} \equiv \langle f(n) \mid 0 \leq n < N \rangle \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{l-1} 10^{(\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))(il+j)} \overset{8.1.4}{\equiv} \\ \prod_{n=0}^{\infty} 10^{(\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f))(n)} = \langle \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes f) \rangle\end{gathered}$$

($\impliedby$) If $\sigma \equiv \langle \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \rangle$, then by Lemma 16.9 we have $\langle f \rangle \geq \langle \vec{\beta} \otimes (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)) \rangle \equiv \sigma$ thus by transitivity $\langle f \rangle \geq \sigma$. $\square$

Chapter VI

Deep results

17 $[\langle n^2 \rangle]$ is an atom

In this section, we proof the fact that the degree $[\langle n^2 \rangle]$ is an atom. To do this, we show some corollaries of Theorem 16.10.

Corollary 17.1 ([8])**.** Let $f \in \circlearrowleft$ and $\langle f \rangle \geq \sigma$ with $\sigma \not\equiv \mathbf{0}$, then we have $\sigma \geq \langle [b] \oplus ((\vec{\alpha}) \otimes \mathcal{S}^k(f)) \rangle$ for some $k \geq 0$, a non-constant weight $\vec{\alpha}$ and an integer b.

Proof. For this proof we refer to [8]. $\square$

Proposition 17.2 ([8])**.** If P_k is a polynomial of degree $k \geq 0$ with non-negative integer coefficients and $\langle P_k \rangle \geq \sigma$ with $\sigma \not\equiv \mathbf{0}$, then $\sigma \geq \langle P'_k \rangle$ for some polynomial P'_k of the same degree k with non-negative integer coefficients.

Proof. By Corollary 17.1 we have that

$$\sigma \geq \langle [b] \oplus ((\vec{\alpha}) \otimes \mathcal{S}^c(P_k)) \rangle$$

for some $c > 0,\ b \in \mathbb{Z}$ and a non-constant weight $\vec{\alpha} = [a_0, a_1, ..., a_{l-1}]$ with $l > 0$. We can apply the definition of the mass product and weight displacement:

$$([b] \oplus ((\vec{\alpha}) \otimes \mathcal{S}^c(P_k))) = n \mapsto ([b] \oplus ((\vec{\alpha}) \otimes \mathcal{S}^c(P_k)(n))) =$$

$$n \mapsto ([b] \oplus ((\vec{\alpha}) \otimes P_k(c+n))) = n \mapsto ([b] \oplus (\sum_{j=0}^{l-1} a_j * P_k(c+nl+j))) =$$

$$n \mapsto (b + \sum_{j=0}^{l-1} a_j {*} P_k(c+nl+j))$$

it is clear that the last equation is a polynomial of degree k with non-negative integer coefficients. $\square$

Corollary 17.3 ([8])**.** Let $f : \mathbb{N} \to \mathbb{N}$ and $a, b \in \mathbb{N}$, then $\langle f(n) \rangle \geq \langle af(2n) + bf(2n+1) \rangle$

Proof. This follows from Lemma 16.9 where we take $\vec{\beta} := [0]$ and $\vec{\boldsymbol{\alpha}} := ([a],[b])$ and $k := 0$.

$$
\begin{gathered}
\langle n \mapsto f(n)\rangle \geq \\
\langle n \mapsto (\vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(f)))\rangle = \\
\langle n \mapsto ([0] \oplus (([a],[b]) \otimes \mathcal{S}^0(f)))\rangle = \\
\langle n \mapsto (([a],[b]) \otimes f)\rangle = \\
\langle n \mapsto af(2n) + bf(2n+1)\rangle = \\
\langle af(2n) + bf(2n+1)\rangle
\end{gathered}
$$

□

Theorem 17.4 ([8])**.** The degree $[\langle n^2\rangle]$ is an atom.

Proof. Let $\langle n^2\rangle \geq \sigma$, with $\sigma \not\equiv \mathbf{0}$. By Proposition 17.2 we know that $\sigma \geq \langle n \mapsto an^2 + bn + c\rangle$ for some $a > 0$ and $b, c \geq 0$.

Without loss of generality, assume that $2a \geq b$. If not choose, $d > 0$ such that $2ad \geq b$ and note that $\langle an^2 + bn + c\rangle \overset{9.5.4}{\geq} \langle ad^2n^2 + bdn + c\rangle = \langle a'n^2 + b'n + c\rangle$ where $a' = ad^2$ and $b' = bd$.

We now derive:

$$
\begin{gathered}
\langle an^2 + bn + c\rangle \overset{9.5.3}{\equiv} \langle an^2 + bn\rangle \overset{9.5.1}{\equiv} \\
\langle a(n+1)^2 + b(n+1)\rangle = \langle an^2 + 2na + a + bn + b\rangle \overset{9.5.3}{\equiv} \\
\langle an^2 + (2a+b)n\rangle =: \langle f(n)\rangle
\end{gathered}
$$

We define $f(n) = an^2 + (2a+b)n$. Note that we have that $2a + b \geq 0$. By Corollary 17.3 we can see that:

$$
\begin{gathered}
\langle f(n)\rangle \overset{17.3}{\geq} \langle b(f(2n)) + (2a-b)(f(2n+1))\rangle = \\
\langle b(a(2n)^2 + (2a+b)2n) + (2a-b)(a(2n+1)^2 + (2a+b)(2n+1))\rangle = \\
\langle 8a^2n^2 + 16a^2n + 6a^2 - ab - b^2\rangle \overset{9.5.3}{\equiv} \\
\langle 8a^2n^2 + 16a^2n\rangle \overset{9.5.3}{\equiv} \langle 8a^2(n+1)^2\rangle \overset{9.5.2}{\equiv} \\
\langle (n+1)^2\rangle \overset{9.5.1}{\equiv} \langle n^2\rangle
\end{gathered}
$$

This shows that every transduct of $\langle n^2\rangle$ is either equivalent to $\mathbf{0}$ or can be transduced back to $\langle n^2\rangle$. □

18 Countably many atoms

In the previous section, we have seen that the degree of $\langle n^2 \rangle$ is an atom (Theorem 17.4). We also stated that the degree of the stream $\langle n \rangle$ is an atom (Theorem 11.2) as shown in [1]. In this section, we state some results from [12]. This completes the current state of the research concerning atom degrees.

A surprising result is that, although the degrees of $\langle n \rangle$ and $\langle n^2 \rangle$ are atoms, the degree of $\langle n^3 \rangle$ is not. In fact, the degree of $\langle n^k \rangle$ is not an atom for each $k \geq 3$.

Theorem 18.1 ([12]). For $k \geq 3$, we have that $[\langle n^k \rangle]$ is not an atom.

We will quickly highlight the main result of [12]. Namely, that we can find countably many atom degrees!

Theorem 18.2 ([12]). Let $k \geq 1$. Then there exists a polynomial P_k of degree k such that $[\langle P_k \rangle]$ is an atom.

For the proofs of both statements, we refer to the paper *Degrees of Infinite Words, Polynomials and Atoms* [12].

19 Suprema

As previously mentioned, not every set of transducer degrees has a supremum. This was shown in [9] by giving an example of a set of two degrees that does not have a supremum.

Definition 19.1. A set $S \subseteq (\mathbf{2}^{\mathbb{N}}/\equiv)$ of transducer degrees has a *least upper bound* or a *supremum* $[s] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ if, for each upper bound $[u] \in (\mathbf{2}^{\mathbb{N}}/\equiv)$ of S we have $[u] \geq [s]$.

We first give a definition of an equivalence relation on natural functions:

Definition 19.2 ([9])**.** Let $f, h : \mathbb{N} \to \mathbb{N}$. Then we define the relation "$\approx$" as follows:

$$f \approx h \iff \exists c_1, c_2 > 0 \exists n_f, n_h \geq 0 \forall n \qquad c_1 h(n_h + n) \leq f(n_f + n) \leq c_2 h(n_h + n)$$

The idea of the next proof is that mass products, weight displacements and shifts as defined in Definitions 9.4 and 16.2 do not change their equivalence by "$\approx$". So if $f \approx h$ then $f \approx \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes \mathcal{S}^k(h))$ for every positive mass $\vec{\boldsymbol{\alpha}}$, every weight $\vec{\beta}$ and each $k \geq 0$.

Theorem 19.3 ([9])**.** The set of degrees $S := \{[\langle 2^{2^n} \rangle], [\langle 3^{3^{3^n}} \rangle]\} \subseteq (\mathbf{2}^{\mathbb{N}}/\equiv)$ does not have a supremum.

Proof. We will give a sketch of why this is the case.

$$t_1 := 2^{2^n} \qquad\qquad t_1 := 3^{3^{3^n}} \tag{19.4}$$

$$u_1 := zip_2(2^{2^n}, 3^{3^{3^n}}) \qquad\qquad u_2 := zip_2(3^{3^{3^n}}, 2^{2^n}) \tag{19.5}$$

$$\tau_1 := \langle t_1 \rangle \qquad\qquad \tau_1 := \langle t_2 \rangle \tag{19.6}$$

$$\mu_1 := \langle zip_2(t_1, t_2) \rangle \qquad\qquad \mu_2 := \langle zip_2(t_1, t_2) \rangle \tag{19.7}$$

One can clearly see that $n \mapsto 2^{2^n} \not\approx n \mapsto 3^{3^{3^n}}$.
By Lemma 10.9 we know that μ_1 and μ_2 are upper bounds of $\{[\tau_1], [\tau_2]\}$.

It was shown in Corollary 12.7 that $t_1, t_2 \in \circlearrowleft$. By Lemma 12.9 we know that $u_1, u_2 \in \circlearrowleft$. We can therefore apply Theorem 16.10 and see that:

$$\mu_1 \geq \theta_1 \implies \theta_1 \equiv \vec{\beta} \oplus (\vec{\boldsymbol{\alpha}} \otimes u_1)) \tag{19.8}$$

$$\mu_2 \geq \theta_2 \implies \theta_2 \equiv \vec{\beta}' \oplus (\vec{\boldsymbol{\alpha}}' \otimes u_2)) \tag{19.9}$$

for some weights $\vec{\beta}, \vec{\beta}'$ and masses $\vec{\boldsymbol{\alpha}}, \vec{\boldsymbol{\alpha}}'$.
Suppose that a supremum degree $[\gamma] \subseteq (\mathbf{2}^{\mathbb{N}}/\equiv)$ exists. By the definition of a supremum we need to have that:

$$\mu_1 \geq \gamma \text{ and } \gamma \geq \tau_1 \wedge \gamma \geq \tau_1 \tag{19.10}$$

$$\mu_2 \geq \gamma \text{ and } \gamma \geq \tau_1 \wedge \gamma \geq \tau_2 \tag{19.11}$$

So we know that $\gamma = \langle g \rangle$ for some spiralling function $g \in \circlearrowleft$ because of equations (19.8) and (19.9). In [9] it was shown that equation (19.10) implies $u_1 \approx g$ and that equation (19.11) implies $u_2 \approx g$ but then by transitivity of $\approx$ we would have that $u_1 \approx u_2$, which contradicts the fact that $u_1 \not\approx u_2$.

$\square$

The above proof shows that not all sets of degrees have a supremum. This does however not say that there do not exist sets of degrees with a supremum. We only know that, in general, not all sets of degrees have a supremum. Whether there are sets that do have a supremum is still an open question.

Chapter VII

Conclusion and future work

Conclusion

In this thesis, we have shown properties of transducer degrees. We showed the concept of function streams as a way to create new streams. Furthermore, we gave the definition of spiralling functions and some of their properties. We gave the definition of zero loops and introduced block loops. We used these loops to give a meaningful structure to the transducts of spiralling function streams. Moreover, we have shown definitions concerning weights and masses and used this to compress the transducts of spiralling function streams to a new form. We used this form to prove that the degree of $\langle n^2 \rangle$ is an atom. We briefly sketched why not all sets of degrees have a supremum. We list the properties that we touched upon in this thesis below.

Some properties

- There exists a bottom degree, namely, the degree of all ultimately periodic sequences.
- Every countable set of degrees has an upper bound.
- The degree of $\langle n \rangle$ is an atom.
- The degree of $\langle n^2 \rangle$ is an atom.
- The degree of $\langle n^k \rangle$ is not atom for $k \geq 3$.
- There are countably many atoms.
- The set of of degrees $S = \{[\langle 2^{2^n} \rangle], [\langle 3^{3^{3^n}} \rangle]\}$ does not have a supremum.

Future Work

There are still a lot of open questions around the topic of transducer degrees. Some that are connected to the concepts explored in this thesis are mentioned below:

1. Is the degree of the Thue-Morse stream an atom?

2. Are there uncountably many atoms?
3. Are there degrees that only transduce to atom degrees or the zero degree?
4. Are there sets of degrees that do have a supremum. (We have seen that this is not the case in general).

The research concerning transducer degrees is still very active. The proof that there are degrees without a supremum was only published online in december of 2019. We recommend to keep an eye on the website of Jörg Endrullis to be informed about the latest discoveries in this field [6].

References

[1] J. Endrullis, D. Hendriks, and J. W. Klop. Degrees of Streams, journal, 2011. Proceedings of the Leiden Numeration Conference 2010.

[2] H. B. Enderton. Elements of recursion theory. In J. Barwise, editor, *HANDBOOK OF MATHEMATICAL LOGIC*. Volume 90, Studies in Logic and the Foundations of Mathematics, pages 527–566. Elsevier, 1977. DOI: `https://doi.org/10.1016/S0049-237X(08)71114-5`. URL: `http://www.sciencedirect.com/science/article/pii/S0049237X08711145`.

[3] L. P. Sasso. Joseph R. Shoenfield. Degrees of unsolvability. North-Holland mathematical studies 2. North-Holland Publishing Company, Amsterdam-London, and American Elsevier Publishing Company, Inc., New York, 1971, VIII 111 pp. *Journal of Symbolic Logic*, 40(3):452–453, 1975. DOI: `10.2307/2272174`.

[4] J. Endrullis, J. W. Klop, A. Saarela, and M. A. Whiteland. Degrees of Transducibility. In *Proc. Conf. on Combinatorics on Words (WORDS 2015)*, volume 9304 of *LNCS*, pages 1–13. Springer, 2015. DOI: `10.1007/978-3-319-23660-5\_1`.

[5] A. Belovs. Some algebraic properties of machine poset of infinite words. en. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 42(3):451–466, 2008. DOI: `10.1051/ita:2008009`. URL: `www.numdam.org/item/ITA_2008__42_3_451_0/`.

[6] Jörg Endrullis. Research: automata. URL: `http://joerg.endrullis.de/research/finite-state-transducers/`.

[7] H. Zantema and W. Bosma. Classifying non-periodic sequences by permutation transducers. In É. Charlier, J. Leroy, and M. Rigo, editors, *Developments in Language Theory*, pages 365–377, Cham. Springer International Publishing, 2017. ISBN: 978-3-319-62809-7.

[8] J. Endrullis, C. Grabmayer, D. Hendriks, and H. Zantema. The Degree of Squares is an Atom. In *Proc. Conf. on Combinatorics on Words (WORDS 2015)*, volume 9304 of *LNCS*, pages 109–121. Springer, 2015. DOI: `10.1007/978-3-319-23660-5\_10`.

[9] J. Endrullis, J. W. Klop, and R. Bakhshi. Transducer Degrees: Atoms, Infima and Suprema. *Acta Informatica*, 57(3-5):727–758, 2020. DOI: `10.1007/s00236-019-00353-7`.

[10] OEIS Foundation Inc. (2020). The on-line encyclopedia of integer sequences. URL: `http://oeis.org/`.

[11] J. Jaffe. A necessary and sufficient pumping lemma for regular languages. *SIGACT News*, 10(2):48–49, July 1978. ISSN: 0163-5700. DOI: `10.1145/990524.990528`. URL: `https://doi.org/10.1145/990524.990528`.

[12] J. Endrullis, J. Karhumäki, J. W. Klop, and A. Saarela. Degrees of Infinite Words, Polynomials and Atoms. In *Proc. Conf. on Developments in Language Theory (DLT 2016)*, volume 9840 of *LNCS*, pages 164–176. Springer, 2016. DOI: `10.1007/978-3-662-53132-7\_14`.