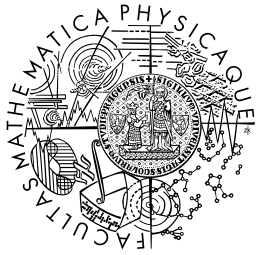


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Tomáš Svoboda

Star height

Department of Algebra

Supervisor of the bachelor thesis:	doc. Štěpán Holub, Ph.D.
Study programme:	Mathematics
Study branch:	Mathematical Methods of Information Security

Prague 2017

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

Prague, 21 July, 2017 Tomáš Svoboda

Title: Star height

Author: Tomáš Svoboda

Department: Department of Algebra

Supervisor: doc. Štěpán Holub, Ph.D., Department of Algebra

Abstract: We present a certain family of languages and show that for those languages infinite hierarchy of star heights exists. The proof was first devised by Dejean and Schützenberger [1]. More recently it was reformulated by Sakarovitch [2], who left some of the parts of the proof the the reader for more careful consideration. This thesis expands on those parts and provides more detailed proofs. We mainly focus on construction of rational expression with the star height of the given language. We also compare the star height and generalised star height and the difference in achieved results for those two similar concepts.

Keywords: star height, infinite hierarchy, automata, rational expressions

Contents

Introduction

Kleene [3], in result known as Kleene's Theorem, shows that automata and expressions correspond to each other and characterise the same class of languages. Eggan [4] refines this result by defining a measure of complexity for both of them: loop complexity for automata and star height for expressions, and by showing they correspond to each other by characterising the same classes of languages. In this thesis we limit ourselves to the presentation of star height, and to the proof, due to Dejean and Schützenberger [1], which states that the star height hierarchy is infinite. They show that for any integer $k \leq 0$, there exists a rational language of star height k over two letter alphabet. We end by touching on another notion, the generalised star height, which may divide the family of rational languages into only two parts.

The determination of the star height of a language turns out to be one of the most difficult problems in automata theory. McNaughton [5] presented first notable result, an algorithm for finding the star height of certain family of languages, so called *pure-group languages*. Hashiguchi first [6] provided an algorithm for deciding whether or not an arbitrary rational language is of star height one and then [7], after six years, an algorithm to determine the star height of any rational language. The algorithm for the general case was not practical, being of non-ELEMENTARY complexity class. Kirsten [8] devised a more efficient algorithm than Hashiguchi's, decidable in $2^{2^{\mathcal{O}(n)}}$ space.

1. Basic concepts

In this chapter we introduce some standard concepts such as words, languages, rational expressions, and automata, but we also make a few key modifications. For example, instead of the classical letters of an alphabet, we use rational expressions as the labels of transitions in automata.

1.1 Words and languages

An *alphabet* is a non-empty, usually finite, set of symbols. Let A be an alphabet. The elements of A, the symbols, are called *letters*, and finite sequences of letters are called *words*. The set of words, sequences of letters of A, is written A^* for reasons that should later become obvious. Word f can therefore be written

$$f = (a_1, a_2, \ldots, a_n),$$

with a_i in A, $1 \leq i \leq n$.

Product

Product on the set of words is equivalent to the operation of concatenation:

$$(a_1, \ldots, a_n) \cdot (b_1, \ldots, b_m) = (a_1, \ldots, a_n, b_1, \ldots, b_m).$$

This operation has a neutral element: the empty sequence or empty word, which is written 1, or 1_{A^*} if there could be some ambiguity. Note, that concatenation is associative, but it is not commutative. The definition of product implies that each word is a product of its letters and therefore can be written

$$f = (a_1, a_2, \ldots, a_n) = (a_1) \cdot (a_2) \cdots (a_n).$$

By identifying the sequence (a) with the letter a and omitting the explicit symbol for product, we are able to write the word f as

$$f = a_1 a_2 \cdots a_n.$$

Similarly for the product of two words f and g we write fg.

Length

The *length* of a word is the length of the sequence of letters the word contains. Let $a_1, a_2, \ldots, a_n$ be letters. Then the length of a word $f = a_1 a_2 \cdots a_n$ is n. It is written $|f|$. For the number of occurrences of the letter a in f we write $|f|_a$. If f is a word in A^* we then have

$$|f| = \sum_{a \in A} |f|_a \, .$$

Factors

Let f, g, h and u be words in A^*. Word g is a *left factor* or *prefix* of f if there exists h such that $f = gh$ and g is a *proper left factor* or *proper prefix* if h is other than the empty word. Word h is a *right factor* or *suffix* of f if there exists g such that $f = gh$ and h is a *proper right factor* or *proper suffix* if g is other than the empty word. Word u is a factor of f if there exist g and h such that $f = guh$ and u is a *proper factor* if g and h are not both equal to the empty word.

Languages

A *language over* A, or *language of* A^*, is any set of words written in the alphabet A. In other words, a language of A^* is a subset of A^*, therefore an element of $\mathbb{P}(A^*)$, the set of all the subsets of A^*. We can thus naturally define for languages all the usual operations on the subsets of a set: union, intersection, complement and difference, with the usual notation.

Word f is *a factor of language* L, if it is a factor of some word g in L.

The product of words extends to a product of languages, which is denoted, as in the case of words, by a dot, $X \cdot Y$, or by simple concatenation, XY:

$$X \cdot Y = \{f \cdot g \mid f \in X\ ,\ g \in Y\}.$$

From this we obtain, by induction on n, the definition of the nth power of X, for all X in $\mathbb{P}(A^*)$:

$$\begin{aligned} X^0 &= \{1_{A^*}\}\ , \\ X^{n+1} &= X \cdot X^n = X^n \cdot X. \end{aligned}$$

Note that even for $\emptyset \in \mathbb{P}(A^*)$ we get $\emptyset^0 = \{1_{A^*}\}$.

The star of a language X is the union of all the powers of X, and is written as X^*:

$$X^* = \bigcup_{n \in \mathbb{N}} X^n.$$

Note that the notation X^* is somewhat improper because in this context X^* does not denote the set of all words over X but the set of products of elements of X. Nevertheless, if one takes $X = A$, the impropriety vanishes: the set of products of elements of an alphabet A is indeed equal to the set of all words generated by A.

1.2 Rational expressions

Let A be an alphabet and let $\{\mathsf{0}, \mathsf{1}, +, \cdot, *\}$ be five function symbols.

Definition. *A rational expression over* A is a formula obtained inductively from the letters of A and the symbols $\{\mathsf{0}, \mathsf{1}, +, \cdot, *\}$ by the following process:

(i) $\mathsf{0}, \mathsf{1}$, and a, for a in A, are rational expressions,

(ii) if E and F are rational expressions, then $(\mathsf{E}+\mathsf{F})$, $(\mathsf{E}\cdot\mathsf{F})$, and (E^*) are rational expressions.

We write $\mathsf{RatE}A^*$ for the set of rational expressions over A. □

To be able to simplify the notation, we can think of the function symbols as symbols representing operations and we can omit the parentheses in the rational expressions if we specify the order of precedence for $+$, $\cdot$, and $*$. We let $*$ take precedence over $\cdot$, which in turn takes precedence over $+$. As with letters in words, we omit the symbol $\cdot$. With this convention we write, for example,

$$\begin{array}{lll} ab + ba & \text{for} & ((a \cdot b) + (b \cdot a)), \text{ or} \\ (a + b\,(ab^*a)^*b)^* & \text{for} & ((a + (b \cdot ((((a \cdot (b^*)) \cdot a)^*) \cdot b)))^*). \end{array}$$

Definition. To each rational expression E in $\mathsf{RatE}A^*$ we assign a language of A^*, written $L[\mathsf{E}]$, and defined inductively:

$$L[\mathsf{0}] = \emptyset\,, \quad L[\mathsf{1}] = \{1_{A^*}\}\,, \quad \text{and} \quad L[a] = \{a\} \quad \text{for all } a \text{ in } A$$

for the atomic rational expressions and then for the composite rational expressions:

$$L[\mathsf{E} + \mathsf{F}] = L[\mathsf{E}] \cup L[\mathsf{F}]\,, \quad L[\mathsf{E} \cdot \mathsf{F}] = L[\mathsf{E}] \cdot L[\mathsf{E}]\,, \quad L[\mathsf{E}^*] = (L[\mathsf{E}])^*.$$

We say that E *denotes* the language $L[\mathsf{E}]$. □

Note that the symbol $*$ is used in two different contexts. Either as a shorthand for the infinite union of all the powers of a language, or, here, as a symbol in a formula of rational expression, in which it, however, directly represents the star of the language the rational expression denotes. This direct translation and the fact that it is a standard notation, we feel, justifies the impropriety of the somewhat excessive usage of the symbol.

We say that a language of A^* is *rational* if and only if it is denoted by a rational expression over A.

Two rational expressions are *equivalent* if they denote the same language. That means E and F are equivalent if $L[\mathsf{E}] = L[\mathsf{F}]$ and we write it as $\mathsf{E} \equiv \mathsf{F}$.

Identities

Let us state a few identities of rational expressions that will be often used without explicit mention.

Lemma 1.

$$(\mathsf{E} + \mathsf{F}) + \mathsf{G} \equiv \mathsf{E} + (\mathsf{F} + \mathsf{G})\,, \quad \textit{and} \quad (\mathsf{E} \cdot \mathsf{F}) \cdot \mathsf{G} \equiv \mathsf{E} \cdot (\mathsf{F} \cdot \mathsf{G})\,,$$

$$\mathsf{E} + \mathsf{F} \equiv \mathsf{F} + \mathsf{E}\,,$$

$$\mathsf{E} + \mathsf{0} \equiv \mathsf{0} + \mathsf{E} \equiv \mathsf{E}\,, \quad \mathsf{E} \cdot \mathsf{0} \equiv \mathsf{0} \cdot \mathsf{E} \equiv \mathsf{0}\,, \quad \mathsf{E} \cdot \mathsf{1} \equiv \mathsf{1} \cdot \mathsf{E} \equiv \mathsf{E}\,,$$

$$\mathsf{E} \cdot (\mathsf{F} + \mathsf{G}) \equiv \mathsf{E} \cdot \mathsf{F} + \mathsf{E} \cdot \mathsf{G}\,, \quad \textit{and} \quad (\mathsf{E} + \mathsf{F}) \cdot \mathsf{G} \equiv \mathsf{E} \cdot \mathsf{G} + \mathsf{F} \cdot \mathsf{G}\,.$$

Proof.

$$L[(\mathsf{E}+\mathsf{F})+\mathsf{G}] = L[\mathsf{E}+\mathsf{F}] \cup L[\mathsf{G}] = L[\mathsf{E}] \cup L[\mathsf{F}] \cup L[\mathsf{G}] = L[\mathsf{E}] \cup L[\mathsf{F}+\mathsf{G}]$$
$$= L[\mathsf{E}+(\mathsf{F}+\mathsf{G})] \ ,$$

$$L[(\mathsf{E}\cdot\mathsf{F})\cdot\mathsf{G}] = L[\mathsf{E}\cdot\mathsf{F}]\cdot L[\mathsf{G}] = L[\mathsf{E}]\cdot L[\mathsf{F}]\cdot L[\mathsf{G}] = L[\mathsf{E}]\cdot L[\mathsf{F}\cdot\mathsf{G}]$$
$$= L[\mathsf{E}\cdot(\mathsf{F}\cdot\mathsf{G})] \ ,$$

$$L[\mathsf{E}+\mathsf{F}] = L[\mathsf{E}] \cup L[\mathsf{F}] = L[\mathsf{F}] \cup L[\mathsf{E}] = L[\mathsf{F}+\mathsf{E}] \ ,$$

$$L[\mathsf{E}+\mathsf{0}] = L[\mathsf{E}] \cup L[\mathsf{0}] = L[\mathsf{E}] = L[\mathsf{0}] \cup L[\mathsf{E}] = L[\mathsf{0}+\mathsf{E}] \ ,$$

$$L[\mathsf{E}\cdot\mathsf{0}] = L[\mathsf{E}]\cdot L[\mathsf{0}] = L[\mathsf{0}] = L[\mathsf{0}]\cdot L[\mathsf{E}] = L[\mathsf{0}\cdot\mathsf{E}] \ ,$$

$$L[\mathsf{E}\cdot\mathsf{1}] = L[\mathsf{E}]\cdot L[\mathsf{1}] = L[\mathsf{E}] = L[\mathsf{1}]\cdot L[\mathsf{E}] = L[\mathsf{1}\cdot\mathsf{E}] \ ,$$

$$L[\mathsf{E}\cdot(\mathsf{F}+\mathsf{G})] = L[\mathsf{E}]\cdot L[\mathsf{F}+\mathsf{G}] = L[\mathsf{E}]\cdot(L[\mathsf{F}]\cup L[\mathsf{G}]) = L[\mathsf{E}]\cdot L[\mathsf{F}]\cup L[\mathsf{E}]\cdot L[\mathsf{G}]$$
$$= L[\mathsf{E}\cdot\mathsf{F}]\cup L[\mathsf{E}\cdot\mathsf{G}] = L[\mathsf{E}\cdot\mathsf{F}+\mathsf{E}\cdot\mathsf{G}] \ ,$$

$$L[(\mathsf{E}+\mathsf{F})\cdot\mathsf{G}] = L[\mathsf{E}+\mathsf{F}]\cdot L[\mathsf{G}] = (L[\mathsf{E}]\cup L[\mathsf{F}])\cdot L[\mathsf{G}] = L[\mathsf{E}]\cdot L[\mathsf{G}]\cup L[\mathsf{F}]\cdot L[\mathsf{G}]$$
$$= L[\mathsf{E}\cdot\mathsf{G}]\cup L[\mathsf{F}\cdot\mathsf{G}] = L[\mathsf{E}\cdot\mathsf{G}+\mathsf{F}\cdot\mathsf{G}] \ .$$

■

We also have

$$\mathsf{0}^* \equiv \mathsf{1} \ ,$$

since

$$L[\mathsf{0}^*] = \emptyset^* = \{1\} = L[\mathsf{1}] \ .$$

1.3 Star height

The symbol $*$ defined for rational expressions is the only one that takes rational expression E denoting a finite language and gives rational expression denoting an infinite language, E^*. Hence the idea of measuring the complexity of expressions by counting the number of nested uses of this symbol, a number which is called the *star height*, which we will write $\mathsf{h}[\mathsf{E}]$, for $\mathsf{E} \in \mathsf{Rat}\mathsf{E}A^*$, and which is defined by induction:

if $\mathsf{E} = \mathsf{0}, \mathsf{E} = \mathsf{1}$ or $\mathsf{E} = a$, for $a \in A$,	$\mathsf{h}[\mathsf{E}] = 0$,
if $\mathsf{E} = \mathsf{F}+\mathsf{G}$ or $\mathsf{E} = \mathsf{F}\cdot\mathsf{G}$,	$\mathsf{h}[\mathsf{E}] = \max(\mathsf{h}[\mathsf{F}], \mathsf{h}[\mathsf{G}])$,
if $\mathsf{E} = \mathsf{F}^*$,	$\mathsf{h}[\mathsf{E}] = 1 + \mathsf{h}[\mathsf{F}]$.

The star height of rational language L over A^*, written $\mathsf{h}[L]$, is the minimum of the star heights of the rational expressions that denote L:

$$\mathsf{h}[L] = \min\{\mathsf{h}[\mathsf{E}] \mid \mathsf{E} \in \mathsf{Rat}\mathsf{E}A^* : L = L[\mathsf{E}]\} \ .$$

Lemma 2. *Every language L over A with star height h is denoted by a finite sum of rational expressions:*

$$L = L[\mathsf{E}_1 + \cdots + \mathsf{E}_n] \ ,$$

where each E_j is a product of the form:

$$\mathsf{E}_j = u_0 \mathsf{F}_1^* u_1 \cdots u_{m-1} \mathsf{F}_m^* u_m,$$

where each $u_k, 0 \leq k \leq m$, is a word in A^* and each $\mathsf{F}_k, 1 \leq k \leq m$, is a rational expression over A of star height less then or equal to $h-1$.

Proof. By induction on h, follows from the fact that product distributes over union. ■

1.4 Automata

An automaton is a *directed graph* which is *labelled* with rational expressions over an alphabet, and in which two subsets of vertices are distinguished.

Definition. An automaton $\mathcal{A}$ is specified by giving the following elements:

(i) a non-empty set Q, called the set of *states* of $\mathcal{A}$,

(ii) a set A, also non-empty, called the *(input) alphabet* of $\mathcal{A}$,

(iii) a subset I of Q, called the set of *initial states*, of $\mathcal{A}$,

(iv) a subset T of Q, called the set of *final states* of $\mathcal{A}$,

(v) a subset E of $Q \times \mathsf{Rat}\mathsf{E}A^* \times Q$, called the set of *transitions* of $\mathcal{A}$.

We write $\mathcal{A} = \langle Q, A, E, I, T \rangle$ and we say that $\mathcal{A}$ is *an automaton over* A. □

Let A be a finite alphabet. We call an automaton over A *finite* if set its states is finite.

If $e = (p, \mathsf{E}, q)$ is a transition of $\mathcal{A}$, that is, if e is in E, we say that E is the *label* of e and we will write $p \xrightarrow{\mathsf{E}} q$, or $p \xrightarrow[\mathcal{A}]{\mathsf{E}} q$ where it might be ambiguous which automaton we are considering. We also say that p is the *source* and q the *destination* of the transition e. Transition is a *loop*, if its source and destination are the same state. When transition has label $\mathsf{1}$, it is called *spontaneous*.

Note that we can always assume that there is, between each pair of states of an automaton, *at most one transition*. This is because we chose rational expressions as the labels of transitions, rather than the usual letters of A. We make this generalisation to be able to represent any automaton by a single transition, as was shown to by possible by Kleene [3]. Similar generalisation was first introduced by Brzozowski and McCluskey [9].

When appropriate, we can even assume that there is, between each pair of states of an automaton, *exactly one transition*, since if for some pair of states, there is no transition in the automaton, we can add a transition labelled with $\mathsf{0}$.

A *computation* in $\mathcal{A}$ is a sequence of transitions where the source of each transition is the destination of the previous one, which can be written as:

$$p_0 \xrightarrow{\mathsf{E}_1} p_1 \xrightarrow{\mathsf{E}_2} p_2 \xrightarrow{\mathsf{E}_3} \cdots \xrightarrow{\mathsf{E}_{n-1}} p_{n-1} \xrightarrow{\mathsf{E}_n} p_n,$$

or

$$p_0 \xrightarrow{\mathsf{E}_1\mathsf{E}_2\cdots\mathsf{E}_n} p_n.$$

We say that *computation is in* $\mathcal{A}$ if every transition of the computation is in $\mathcal{A}$. The state p_0 is the *source* of the computation c, and p_n its *destination*. The *length* of the computation c is n, the number of transitions which make up c. The *label* of c is the product of the labels of the transitions of c. In the above case, the label of c is $\mathsf{E}_1\mathsf{E}_2\cdots\mathsf{E}_n$.

A computation in $\mathcal{A}$ is *successful* if its source is an initial state and its destination is a final state. A word in A^* is called *accepted* or *recognised* by $\mathcal{A}$ if it is in a language denoted by a label of a successful computation in $\mathcal{A}$.

Definition. The language accepted, or recognised by $\mathcal{A}$, written $L(\mathcal{A})$, is the set of words *accepted* (or *recognised*) by $\mathcal{A}$:

$$L(\mathcal{A}) = \{f \in A^* \mid \exists\, i \in I\ ,\ \exists\, t \in T :\ i \xrightarrow[\mathcal{A}]{\mathsf{E}} t\ \wedge\ f \in L[\mathsf{E}]\}.$$

□

Two automata are *equivalent* if they recognise the same language. If L is a language, and a finite automaton $\mathcal{A}$ exists such that $L = L(\mathcal{A})$, we call L *recognisable*.

Lemma 3. *Let L be a recognisable language over A. There exists $N \in \mathbb{N}$ such that for every word f in L and every factorisation of the form $f = uv_1v_2\cdots v_Nw$, where every v_i is a non-empty word, there is a pair (j,k) of indices, $0 \le j < k \le N$, that*

$$uv_1v_2\cdots v_j(v_{j+1}\cdots v_k)^*v_{k+1}\cdots v_Nw \subseteq L.$$

Proof. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be an automaton that recognises L. Set $N = |Q|$, meaning N is a size of the set Q. A computation that accepts f can be written

$$i \xrightarrow{\mathsf{E}} q_0 \xrightarrow{\mathsf{F}_1} q_1 \xrightarrow{\mathsf{F}_2} q_2 \xrightarrow{\mathsf{F}_3} \cdots \xrightarrow{\mathsf{F}_{N-1}} q_{N-1} \xrightarrow{\mathsf{F}_N} q_N \xrightarrow{\mathsf{G}} t,$$

where $u \in L[\mathsf{E}]$, $v_i \in L[\mathsf{F}_i]$, for $1 \le i \le N$, and $w \in L[\mathsf{G}]$. The $N+1$ states q_i cannot all be distinct, and at least two, say q_j and q_k, are equal to the same state p. The computation can therefore be written

$$i \xrightarrow{\mathsf{E}} q_0 \xrightarrow{\mathsf{F}_1\mathsf{F}_2\cdots\mathsf{F}_j} p \xrightarrow{\mathsf{F}_{j+1}\cdots\mathsf{F}_k} p \xrightarrow{\mathsf{F}_{k+1}\cdots\mathsf{F}_N} q_N \xrightarrow{\mathsf{G}} t,$$

where $L[\mathsf{F}_{j+1}\cdots\mathsf{F}_k]$ contains the non-empty word $v_{j+1}\cdots v_k$. Hence, for every non-negative integer n,

$$i \xrightarrow{\mathsf{E}} q_0 \xrightarrow{\mathsf{F}_1\mathsf{F}_2\cdots\mathsf{F}_j} p \xrightarrow{(\mathsf{F}_{j+1}\cdots\mathsf{F}_k)^n} p \xrightarrow{\mathsf{F}_{k+1}\cdots\mathsf{F}_N} q_N \xrightarrow{\mathsf{G}} t,$$

is a successful computation of $\mathcal{A}$ and $uv_1v_2\cdots v_j(v_{j+1}\cdots v_k)^nv_{k+1}\cdots v_Nw$ is accepted by $\mathcal{A}$. ■

Let $\mathbb{Z}_n$ be finite cyclic group of order n. When using $\mathbb{Z}_n$ as a set of states of an automaton, we consider integers $\{0, 1, 2, \ldots, n-1\}$ to be the underlying set of the group and we utilise the group operations modulo n when enumerating the states.

Definition. Automaton $\langle \mathbb{Z}_n, \{a, b\}, E, \{0\}, \{0\} \rangle$ is called *ring automaton* $\mathcal{R}(n)$, if for each state $z \in \mathbb{Z}_n$ there are exactly two transitions with z as a source and they are $(z, a, z+1)$ and $(z, b, z-1)$. □

Note that if state is a member of $\mathbb{Z}_n$ we often denote it by some addition. For example, let s, r by states of $\mathcal{R}(n)$. Then $s \xrightarrow{\mathsf{E}} s+r$ denotes transition $(s, \mathsf{E}, s+r)$, and $s \xrightarrow{\mathsf{E}} s+r \xrightarrow{\mathsf{F}} s+2$ denotes computation consisting of transitions $(s, \mathsf{E}, s+r)$ and $(s+r, \mathsf{F}, s+2)$.

Example. Figure 1.1 shows ring automaton $\mathcal{R}(8)$.

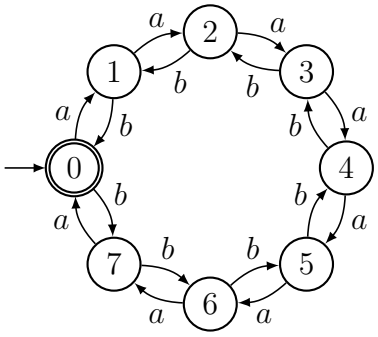


Figure 1.1: Automaton $\mathcal{R}(8)$

□

Lemma 4. *Let $\mathcal{R}(n)$ be a ring automaton. For any m, r, and t in $\mathbb{Z}_n$, if the computation $r \xrightarrow{\mathsf{H}} t$ is in $\mathcal{R}(n)$ then $r + m \xrightarrow{\mathsf{H}} t + m$ is also in $\mathcal{R}(n)$.*

Proof. By induction on m. Let $m = 0$. Since state $r + 0$ is the same as r and $s + 0$ is the same as s, the proposition holds. Given the induction hypothesis for some $m - 1$, let the computation $r \xrightarrow{\mathsf{H}} t$ be written as a sequence of transitions:

$$s_0 \xrightarrow{\mathsf{X}_1} s_1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{k-1}} s_{k-1} \xrightarrow{\mathsf{X}_k} s_k \,,$$

where $s_0 = r\,,\ s_k = t$, and each X_i is either a letter a or b. Then computation

$$s_0 + m - 1 \xrightarrow{\mathsf{X}_1} s_1 + m - 1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{k-1}} s_{k-1} + m - 1 \xrightarrow{\mathsf{X}_k} s_k + m - 1$$

is in $\mathcal{R}(n)$. For any state $p \in \mathcal{R}(n)$, transitions $(p, a, p+1)$ and $(p, b, p-1)$ are in $\mathcal{R}(n)$, thus, if $(s_i + m - 1, \mathsf{X}_\mathsf{i}, s_i + m - 1)$ is in $\mathcal{R}(n)$, so is $(s_i + m, \mathsf{X}_\mathsf{i}, s_i + m)$, for $0 \leq i \leq k$. Therefore

$$s_0 + m \xrightarrow{\mathsf{X}_1} s_1 + m \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{k-1}} s_{k-1} + m \xrightarrow{\mathsf{X}_k} s_k + m \,,$$

which is the computation $r + m \xrightarrow{\mathsf{H}} t + m$, is also in $\mathcal{R}(n)$. ■

State removal algorithm

Brzozowski and McCluskey [9] used generalised forms of automata to show that there is a simple algorithm that takes a finite automaton and returns a rational expression denoting the same language that is accepted by the automaton. This algorithm is used in Lemma 8 to find a rational expression denoting the language recognised by a ring automaton.

Lemma 5. *Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be an automaton with n states which has at least one state that is neither initial nor accepting, that means $Q \setminus (I \cup T) \neq \emptyset$. An automaton $\mathcal{B}$ with $n-1$ states exists such that $L(\mathcal{B}) = L(\mathcal{A})$.*

Proof. Let q be a state in $Q \setminus (I \cup T)$. Next, we consider the transitions

$$(p_1, \mathsf{E}_1, q), \ldots, (p_k, \mathsf{E}_k, q), (q, \mathsf{G}_1, r_1), \ldots, (q, \mathsf{G}_l, r_l), \text{ and } (q, \mathsf{F}, q),$$

where each $p_h \neq q$, and $r_j \neq q$. These are all the different transitions of $\mathcal{A}$ with q as a destination, source, or both respectively. Note that F may be $\mathsf{1}$, since every state has at least spontaneous loop. To create the automaton $\mathcal{B}$, we remove all the mentioned transitions and the state q, and for each pair of states (p_h, r_j), $1 \leq h \leq k$ and $1 \leq j \leq l$, we replace the transition (p_h, H, r_j) with $(p_h, \mathsf{H} + \mathsf{E}_h\mathsf{F}^*\mathsf{G}_j, r_j)$. Note that H may be $\mathsf{0}$ for some state pairs.

Now we verify that $L(\mathcal{B}) = L(\mathcal{A})$. Let f be a word accepted by $\mathcal{A}$. A computation that accepts f can be written as

$$i \xrightarrow{\mathsf{A}_1} s_1 \xrightarrow{\mathsf{A}_2} s_2 \xrightarrow{\mathsf{A}_3} \cdots \xrightarrow{\mathsf{A}_{N-1}} s_{N-1} \xrightarrow{\mathsf{A}_N} t.$$

If some $s_M = q$, there are non-negative integers K and L such that

$$s_{M-K} = s_{M-K+1} = \cdots = s_{M-1} = s_M = s_{M+1} = \cdots = s_{M+L-1} = s_{M+L} = q,$$

and the computation accepting f is

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_{M-K-1}} p_h \xrightarrow{\mathsf{E}_h} q \xrightarrow{\overbrace{\mathsf{F}\cdots\mathsf{F}}^{K+L \text{ times}}} q \xrightarrow{\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+L+2}} \cdots \xrightarrow{\mathsf{A}_N} t,$$

therefore a factorisation of f exists such that $f = u_1u_2v_1\cdots v_{K+L}w_1w_2$, where $u_1 \in L[\mathsf{A}_1\cdots\mathsf{A}_{M-K-1}], u_2 \in L[\mathsf{E}_h], v_1, \ldots, v_{K+L} \in L[\mathsf{F}], w_1 \in L[\mathsf{G}_j]$, and $w_2 \in L[\mathsf{A}_{M+L+2}\cdots\mathsf{A}_N]$. It follows that, for every non-negative integer J and any words $g_1, \ldots, g_J \in L[\mathsf{F}]$, the word $u_1u_2g_1\cdots g_Jw_1w_2$ is also in $L(\mathcal{A})$. Specifically it is accepted by computations:

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_{M-K-1}} p_h \xrightarrow{\mathsf{E}_h} q \xrightarrow{\overbrace{\mathsf{F}\cdots\mathsf{F}}^{J \text{ times}}} q \xrightarrow{\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+L+2}} \cdots \xrightarrow{\mathsf{A}_N} t\ ,$$

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_{M-K-1}} p_h \xrightarrow{\mathsf{E}_h} q \xrightarrow{\mathsf{F}^*} q \xrightarrow{\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+L+2}} \cdots \xrightarrow{\mathsf{A}_N} t\ ,$$

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_{M-K-1}} p_h \xrightarrow{\mathsf{E}_h\mathsf{F}^*\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+L+2}} \cdots \xrightarrow{\mathsf{A}_N} t,$$

but also by

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_{M-1}} p_h \xrightarrow{\mathsf{H}+\mathsf{E}_h\mathsf{F}^*\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+L+2}} \cdots \xrightarrow{\mathsf{A}_N} t,$$

which is a successful computation is $\mathcal{B}$ and therefore f is accepted by $\mathcal{B}$.

Otherwise, if every state $s_M \neq q$, but there are some states $s_M = p_h$ and $s_{M+1} = r_j$, then f is accepted by a successful computation:

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_M} p_h \xrightarrow{\mathsf{H}} r_j \xrightarrow{\mathsf{A}_{M+2}} \cdots \xrightarrow{\mathsf{A}_N} t,$$

therefore f is also accepted by $\mathcal{B}$, since

$$i \xrightarrow{\mathsf{A}_1} \cdots \xrightarrow{\mathsf{A}_M} p_h \xrightarrow{\mathsf{H}+\mathsf{E}_h\mathsf{F}^*\mathsf{G}_j} r_j \xrightarrow{\mathsf{A}_{M+2}} \cdots \xrightarrow{\mathsf{A}_N} t$$

is a successful computation in $\mathcal{B}$.

If neither of the above cases are true, that means every $s_M \neq q$ and if any $s_M = p_h$ then $s_{M+1} \neq r_j$, the word f is also accepted by

$$i \xrightarrow[\mathcal{B}]{\mathsf{A}_1} s_1 \xrightarrow[\mathcal{B}]{\mathsf{A}_2} s_2 \xrightarrow[\mathcal{B}]{\mathsf{A}_3} \cdots \xrightarrow[\mathcal{B}]{\mathsf{A}_{N-1}} s_{N-1} \xrightarrow[\mathcal{B}]{\mathsf{A}_N} t,$$

which is successful computation in $\mathcal{B}$, therefore $f \in L[\mathcal{B}]$.

This shows that every word accepted by $\mathcal{A}$ is also accepted by $\mathcal{B}$. ■

The process of removing the state described in the Proof of Lemma 5 is called *state removal algorithm*. By iterating on the states of automaton $\mathcal{A} = \langle Q, A, E, I, T\rangle$, applying state removal algorithm each step, we can create an automaton with only $|I \cup T|$ states that accepts $L(\mathcal{A})$. It is important to note that depending on the order of the states chosen to be removed, we may obtain different automata.

2. Eggan's question

In this chapter we present an answer to the question whether there are rational languages with arbitrarily large star heights. The first example of such languages is given by Eggan [4]. Eggan's example uses an alphabet of size $2^n - 1$ for the language with star height n. He therefore asked whether there are some examples of languages over binary alphabets. Answer to that question, which is affirmative, was provided shortly after by Dejean and Schützenberger [1]. Their answer is also reformulated by Sakarovitch [2], who phrased the proof in the form we work with here.

The family of languages provided by Dejean and Schützenberger is the following:

Definition. Let q be a non-negative integer, then W_q is a language, such that

$$W_q = \{f \in \{a,b\}^* \mid |f|_a \equiv |f|_b \pmod{2^q}\}.$$

□

Theorem 6. *The language W_q has star height q.*

The proof has three parts. First, we find an automaton that recognises W_q in Lemma 7. Second, we infer a rational expression of star height q, denoting W_q, in Lemma 8. And lastly, we show that the star height of the language has to be at least q.

2.1 Automaton recognising W_q

Lemma 7. *Language W_q is recognised by an automaton $\mathcal{R}(2^q)$.*

Proof. We show that each word in W_q is accepted by a successful computation in $\mathcal{R}(2^q)$. Let f be a word in W_q. Since f has even length, let k be such that $2k = |f|$. Since $f \in W_q$, $|f|_a - |f|_b = n2^q$ for some integer n.

For $n = 0$, $|f|_a = |f|_b$. We use induction on k to show that for every factorisation $f = uvw$, if f is accepted by $\mathcal{R}(2^q)$, words $uavbw$ and $ubvaw$ are also accepted by $\mathcal{R}(2^q)$. For $k = 0$, empty word is accepted since the initial state is also final and both computations $0 \xrightarrow{a} 1 \xrightarrow{b} 0$, and $0 \xrightarrow{b} 2^q - 1 \xrightarrow{a} 0$ are successful in $\mathcal{R}(2^q)$. Given the induction hypothesis for some $k-1$, every word $f = uvw$ of length $2k-2$ such that $|f|_a = |f|_b$ is accepted by computation

$$0 \xrightarrow{\mathsf{E}} i \xrightarrow{\mathsf{F}} j \xrightarrow{\mathsf{G}} 0\,,$$

where $u \in L[\mathsf{E}], v \in L[\mathsf{F}]$, and $v \in L[\mathsf{G}]$. We need to show that computations

$$0 \xrightarrow{\mathsf{E}} i \xrightarrow{a} i+1 \xrightarrow{\mathsf{F}} j+1 \xrightarrow{b} j \xrightarrow{\mathsf{G}} 0\,, \quad \text{and}$$
$$0 \xrightarrow{\mathsf{E}} i \xrightarrow{b} i-1 \xrightarrow{\mathsf{F}} j-1 \xrightarrow{a} j \xrightarrow{\mathsf{G}} 0\,,$$

are successful. For any two states r and t of $\mathcal{R}(2^q)$, the computation $r \xrightarrow{\mathsf{H}} t$ can be written as a sequence of transitions:

$$r \xrightarrow{\mathsf{X}_1} s_1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1} \xrightarrow{\mathsf{X}_m} t\,,$$

where each X_l is either a letter a or b. Due to Lemma 4, computations

$$r+1 \xrightarrow{\mathsf{X}_1} s_1+1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1}+1 \xrightarrow{\mathsf{X}_m} t+1\,, \quad \text{and}$$
$$r-1 \xrightarrow{\mathsf{X}_1} s_1-1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1}-1 \xrightarrow{\mathsf{X}_m} t-1$$

are also in $\mathcal{R}(2^q)$, and so are

$$r \xrightarrow{a} r+1 \xrightarrow{\mathsf{X}_1} s_1+1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1}+1 \xrightarrow{\mathsf{X}_m} t+1 \xrightarrow{b} t\,, \quad \text{and}$$
$$r \xrightarrow{b} r-1 \xrightarrow{\mathsf{X}_1} s_1-1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1}-1 \xrightarrow{\mathsf{X}_m} t-1 \xrightarrow{a} t\,.$$

If we take computation $i \xrightarrow{\mathsf{F}} j$ in place of $r \xrightarrow{\mathsf{H}} t$, the assertion is proved.

For $n \neq 0$, $|f|_a - |f|_b = n2^q$. Let g be a word such that $|g|_a = |g|_b$, therefore g is accepted by $\mathcal{R}(2^q)$. Computation accepting g can be written as

$$0 \xrightarrow{\mathsf{X}_1} s_1 \xrightarrow{\mathsf{X}_2} \cdots \xrightarrow{\mathsf{X}_{m-1}} s_{m-1} \xrightarrow{\mathsf{X}_m} 0\,,$$

where each X_l is either a letter a or b. Let $\mathsf{Y}_1, \ldots, \mathsf{Y}_{m+1}$ be expressions such that f is in $L[\mathsf{Y}_1\mathsf{X}_1 \cdots \mathsf{Y}_m\mathsf{X}_m\mathsf{Y}_{m+1}]$. Let e_h be in $L[\mathsf{Y}_h]$ for $1 \leq h \leq m+1$.

$$n2^q = |f|_a - |f|_b = |g|_a + \sum_{h=1}^{m+1} |e_h|_a - |g|_b - \sum_{h=1}^{m+1} |e_h|_b = \sum_{h=1}^{m+1} (|e_h|_a - |e_h|_b)\,.$$

By $m+1$ iterations of Lemma 4, it follows that computation

$$0 \xrightarrow{\mathsf{Y}_1\mathsf{X}_1} p_1 + (|e_1|_a - |e_1|_b) \xrightarrow{\mathsf{Y}_2\mathsf{X}_2} p_2 + \sum_{h=1}^{2} (|e_h|_a - |e_h|_b) \xrightarrow{\mathsf{Y}_3\mathsf{X}_3} \cdots$$
$$\cdots \xrightarrow{\mathsf{Y}_{m-1}\mathsf{X}_{m-1}} p_{m-1} + \sum_{h=1}^{m-1} (|e_h|_a - |e_h|_b) \xrightarrow{\mathsf{Y}_m\mathsf{X}_m\mathsf{Y}_{m+1}} \sum_{h=1}^{m+1} (|e_h|_a - |e_h|_b)\,.$$

is in $\mathcal{R}(2^q)$. It is successful computation in $\mathcal{R}(2^q)$ because

$$n2^q = \sum_{h=1}^{m+1} (|e_h|_a - |e_h|_b) \equiv 0 \pmod{2^q}.$$

■

2.2 Rational expression denoting W_q

We have just proved that language W_q is recognised by the ring automaton $\mathcal{R}(2^q)$. We have also shown that the state removal algorithm takes an automaton and finds another one with one less state, that recognises the same language. We combine these results to find an automaton with only one state, that recognises W_q.

Lemma 8. *Language W_q is denoted by a rational expression of star height q.*

Proof. Due to Lemma 7, language W_q is recognised by an automaton $\mathcal{R}(2^q)$. By iterative application of the state removal algorithm, we will obtain an automaton with only one state. First, we set

$$\mathsf{X}_0 = a\,, \qquad \mathsf{Y}_0 = b\,, \qquad \text{and} \qquad \mathsf{Z}_0 = \mathsf{0}\,,$$

then, for every non-negative integer n,

$$\mathsf{X}_{n+1} = \mathsf{X}_n\mathsf{Z}_n^*\mathsf{X}_n\,, \quad \mathsf{Y}_{n+1} = \mathsf{Y}_n\mathsf{Z}_n^*\mathsf{Y}_n\,, \quad \text{and} \quad \mathsf{Z}_{n+1} = \mathsf{Z}_n + \mathsf{X}_n\mathsf{Z}_n^*\mathsf{Y}_n + \mathsf{Y}_n\mathsf{Z}_n^*\mathsf{X}_n\,.$$

Second, we define automaton $\mathcal{A}_k$ as $\mathcal{A}_k = \langle \mathbb{Z}_{2^{q-k}}, A, E, \{0\}, \{0\}\rangle$ such that any state p in $\mathcal{A}_k$ is source of exactly three transitions in $\mathcal{A}_k$:

$$p \xrightarrow{\mathsf{X}_k} p+1\,, \quad p \xrightarrow{\mathsf{Y}_k} p-1\,, \quad \text{and} \quad p \xrightarrow{\mathsf{Z}_k} p\,.$$

By induction on k, we show that W_q, accepted by $\mathcal{R}(2^q)$, is also accepted by $\mathcal{A}_k$.

Let $k = 0$. Note that for each state p in $\mathcal{A}_0$, transitions

$$p \xrightarrow{\mathsf{X}_0} p+1\,, \quad p \xrightarrow{\mathsf{Y}_0} p-1\,, \quad \text{and} \quad p \xrightarrow{\mathsf{Z}_0} p$$

are in $\mathcal{R}(2^q)$. Similarly, for any state r in $\mathcal{R}(2^q)$, transitions

$$r \xrightarrow{a} r+1\,, \quad r \xrightarrow{b} r-1$$

are in $\mathcal{A}_0$. Therefore $\mathcal{R}(2^q)$ is equivalent to $\mathcal{A}_0$.

Given the induction hypothesis for some $k-1$, W_q is accepted by $\mathcal{A}_{k-1}$. By 2^{q-k} iterations of the state removal algorithm an automaton with 2^{q-k} states is obtained. The set of states of $\mathcal{A}_{k-1}$ is $\mathbb{Z}_{2^{q-k+1}}$. Consecutively, in the increasing order, we remove all the odd states of $\mathcal{A}_{k-1}$, that is, states $1, 3, 5, \ldots, 2^{q-k+1}-1$. Let m be the removed odd state. The process of removing the state is technically the same for all the states, but in the interest of clarity, we will differentiate between m being equal to 1, being between 3 and $2^{q-k+1}-3$, and being equal $2^{q-k+1}-1$.

For $m = 1$, the transitions that have 1 as a source, destination, or both are:

$$1 \xrightarrow{\mathsf{X}_{k-1}} 2\,, \quad 1 \xrightarrow{\mathsf{Y}_{k-1}} 0\,, \quad 0 \xrightarrow{\mathsf{X}_{k-1}} 1\,, \quad 2 \xrightarrow{\mathsf{Y}_{k-1}} 1\,, \quad \text{and} \quad 1 \xrightarrow{\mathsf{Z}_{k-1}} 1\,.$$

Those transitions are removed, in accordance with the state removal algorithm. For each pair of states, $(0,0), (0,2), (2,0), (2,2)$, these transitions are replaced:

$$\begin{aligned}
&0 \xrightarrow{\mathsf{Z}_{k-1}} 0 && \text{replaced by} && 0 \xrightarrow{\mathsf{Z}_{k-1} + \mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}} 0\,,\\
&0 \xrightarrow{\mathsf{E}} 2 && \text{replaced by} && 0 \xrightarrow{\mathsf{E} + \mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} 2\,,\\
&2 \xrightarrow{\mathsf{E}} 0 && \text{replaced by} && 2 \xrightarrow{\mathsf{E} + \mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}} 0\,,\\
&2 \xrightarrow{\mathsf{Z}_{k-1}} 2 && \text{replaced by} && 2 \xrightarrow{\mathsf{Z}_{k-1} + \mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} 2\,.
\end{aligned}$$

For $3 \le m \le 2^{q-k+1}-3$, the states $1, 3, \ldots, m-2$ are already removed. The transitions that have m as a source, destination, or both are:

$$m \xrightarrow{\mathsf{X}_{k-1}} m+1\,, \quad m \xrightarrow{\mathsf{Y}_{k-1}} m-1\,, \quad m-1 \xrightarrow{\mathsf{X}_{k-1}} m\,,$$
$$m+1 \xrightarrow{\mathsf{Y}_{k-1}} m\,, \quad \text{and} \quad m \xrightarrow{\mathsf{Z}_{k-1}} m\,.$$

As in the case $m = 1$, those transitions are removed. For each pair of states, $(m-1, m-1), (m-1, m+1), (m+1, m-1), (m+1, m+1)$, these transitions are replaced:

$$m-1 \xrightarrow{\mathsf{Z}_{k-1}+\mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} m-1 \quad \text{replaced by} \quad m-1 \xrightarrow{\mathsf{Z}_k} m-1\,,$$

since $\mathsf{Z}_k = (\mathsf{Z}_{k-1} + \mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}) + \mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}$,

$$\begin{aligned}
m-1 \xrightarrow{\mathsf{E}} m+1 \quad &\text{replaced by} \quad m-1 \xrightarrow{\mathsf{E}+\mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} m+1\,,\\
m+1 \xrightarrow{\mathsf{E}} m-1 \quad &\text{replaced by} \quad m+1 \xrightarrow{\mathsf{E}+\mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}} m-1\,,\\
m+1 \xrightarrow{\mathsf{Z}_{k-1}} m+1 \quad &\text{replaced by} \quad m+1 \xrightarrow{\mathsf{Z}_{k-1}+\mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} m+1.
\end{aligned}$$

Lastly for $m = 2^{q-k+1} - 1$, the states $1, 3, \ldots, 2^{q-k+1} - 3$ are already removed. Since $m = 2^{q-k+1}-1$, the state $m+1$ is $0 \equiv 2^{q-k+1} \pmod{2^{q-k+1}}$. The transitions that have m as a source, destination, or both are:

$$m \xrightarrow{\mathsf{X}_{k-1}} 0\,, \quad m \xrightarrow{\mathsf{Y}_{k-1}} m-1\,, \quad m-1 \xrightarrow{\mathsf{X}_{k-1}} m\,, \quad 0 \xrightarrow{\mathsf{Y}_{k-1}} m\,, \quad \text{and} \quad m \xrightarrow{\mathsf{Z}_{k-1}} m\,.$$

As for every $m \leq 2^{q-k+1} - 3$, these transitions are removed. For each pair of states, $(m-1, m-1), (m-1, 0), (0, m-1), (0, 0)$, these transitions are replaced:

$$\begin{aligned}
m-1 \xrightarrow{\mathsf{Z}_{k-1}+\mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} m-1 \quad &\text{replaced by} \quad m-1 \xrightarrow{\mathsf{Z}_k} m-1\,,\\
m-1 \xrightarrow{\mathsf{E}} 0 \quad &\text{replaced by} \quad m-1 \xrightarrow{\mathsf{E}+\mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{X}_{k-1}} 0\,,\\
0 \xrightarrow{\mathsf{E}} m-1 \quad &\text{replaced by} \quad 0 \xrightarrow{\mathsf{E}+\mathsf{Y}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}} m-1\,,\\
0 \xrightarrow{\mathsf{Z}_{k-1}+\mathsf{X}_{k-1}\mathsf{Z}_{k-1}^*\mathsf{Y}_{k-1}} 0 \quad &\text{replaced by} \quad 0 \xrightarrow{\mathsf{Z}_k} 0\,.
\end{aligned}$$

Thus, after 2^{q-k} odd states removed, we have automaton with 2^{q-k} states, all non-negative even numbers $\leq 2^{q-k+1} - 2$, that accepts W_q. To complete the inductive step, create automaton $\mathcal{A}_k$, we 'rename' each state $2t$ as t, $t \in \{0, 1, 2, \ldots, 2^{q-k} - 1\}$, meaning we replace each state $2t$ with t, each transition $2t \xrightarrow{\mathsf{F}} s$ with $t \xrightarrow{\mathsf{F}} s$, and $s \xrightarrow{\mathsf{F}} 2t$ with $s \xrightarrow{\mathsf{F}} t$, for every state s.

For $k = q - 2$, automaton $\mathcal{A}_2$ has two states. Note that, to conform with the definition of $\mathcal{A}_2$, it should have six distinct transitions,

$$\begin{aligned}
0 \xrightarrow{\mathsf{X}_k} 1\,, \quad 1 \xrightarrow{\mathsf{Y}_k} 0\,, \quad 0 \xrightarrow{\mathsf{Z}_k} 0\,,\\
1 \xrightarrow{\mathsf{X}_k} 0\,, \quad 0 \xrightarrow{\mathsf{Y}_k} 1\,, \quad 1 \xrightarrow{\mathsf{Z}_k} 1\,.
\end{aligned}$$

We can see that the pair of transitions, $0 \xrightarrow{\mathsf{X}_k} 1$ and $0 \xrightarrow{\mathsf{Y}_k} 1$, can be replaced by $0 \xrightarrow{\mathsf{X}_k+\mathsf{Y}_k} 1$, and the pair of transitions, $1 \xrightarrow{\mathsf{X}_k} 0$ and $1 \xrightarrow{\mathsf{Y}_k} 0$ can be replaced by $1 \xrightarrow{\mathsf{X}_k+\mathsf{Y}_k} 0$.

Automaton $\mathcal{A}_2$ is reduced into one state automaton by application of the state removal algorithm on the state 1. Thus creating a loop labelled with $\mathsf{Z}_{q-1} + (\mathsf{X}_{q-1} + \mathsf{Y}_{q-1})\mathsf{Z}_{q-1}^*(\mathsf{X}_{q-1} + \mathsf{Y}_{q-1})$ on the state 0.

Note the following equivalence:

$$\mathsf{Z}_{q-1} + (\mathsf{X}_{q-1} + \mathsf{Y}_{q-1})\mathsf{Z}_{q-1}^*(\mathsf{X}_{q-1} + \mathsf{Y}_{q-1}) \equiv \mathsf{X}_q + \mathsf{Y}_q + \mathsf{Z}_q\,.$$

Therefore

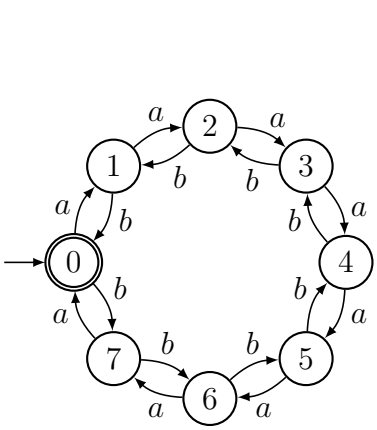
$$W_q = L(\mathcal{A}_q) = L[(\mathsf{Z}_{q-1} + (\mathsf{X}_{q-1} + \mathsf{Y}_{q-1})\mathsf{Z}_{q-1}^*(\mathsf{X}_{q-1} + \mathsf{Y}_{q-1}))^*] = L[(\mathsf{X}_q+\mathsf{Y}_q+\mathsf{Z}_q)^*].$$

By induction on i, we show, that $\mathsf{h}[\mathsf{X}_i] = \mathsf{h}[\mathsf{Z}_i] = \mathsf{h}[\mathsf{Z}_i] = i - 1$. Let $i = 1$. $\mathsf{h}[a^2] = \mathsf{h}[b^2] = \mathsf{h}[ab + ba] = 0$. Given the induction hypothesis for some $i - 1$, $\mathsf{h}[\mathsf{Z}_{i-1}] = i - 1$. Therefore

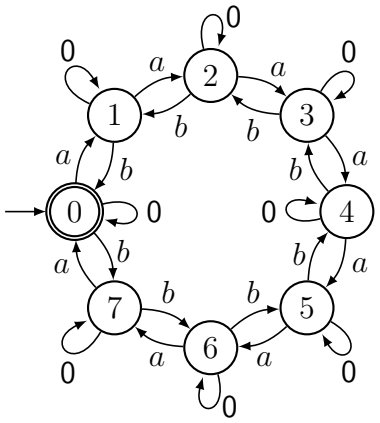
$$\begin{aligned}\mathsf{h}[\mathsf{Z}_{i-1}^*] = i &= \mathsf{h}[\mathsf{X}_{i-1}\mathsf{Z}_{i-1}^*\mathsf{X}_{i-1}] = \mathsf{h}[\mathsf{X}_i] \\ &= \mathsf{h}[\mathsf{Y}_{i-1}\mathsf{Z}_{i-1}^*\mathsf{Y}_{i-1}] = \mathsf{h}[\mathsf{Y}_i] \\ &= \mathsf{h}[\mathsf{Z}_{i-1} + \mathsf{X}_{i-1}\mathsf{Z}_{i-1}^*\mathsf{Y}_{i-1} + \mathsf{Y}_{i-1}\mathsf{Z}_{i-1}^*\mathsf{X}_{i-1}] = \mathsf{h}[\mathsf{Z}_i]\,.\end{aligned}$$

We conclude that $\mathsf{h}[(\mathsf{X}_q + \mathsf{Y}_q + \mathsf{Z}_q)^*] = q$, which proves the lemma. ■

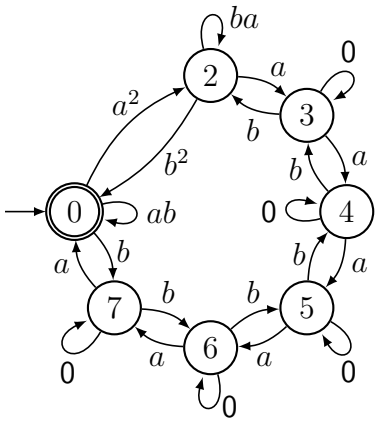
Example. Language W_3 is recognised by $\mathcal{R}(8)$. In Figure 2.1, we show how the rational expression $\mathsf{X}_3 + \mathsf{Y}_3 + \mathsf{Z}_3$ is derived. Then $W_3 = L[(\mathsf{X}_3 + \mathsf{Y}_3 + \mathsf{Z}_3)^*]$. □



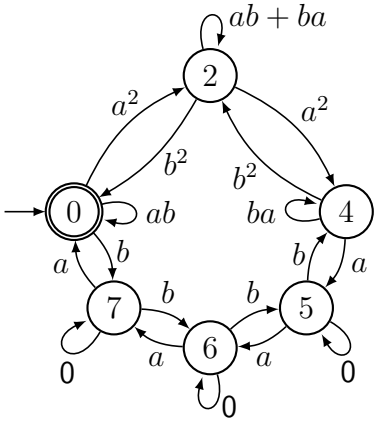
(a) Automaton $\mathcal{R}(8)$



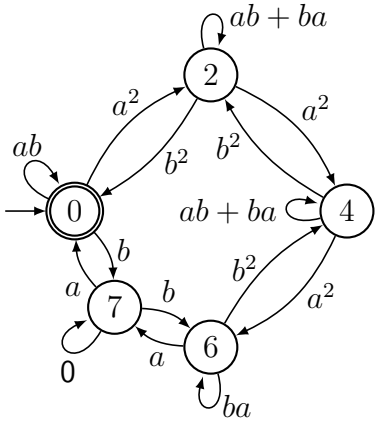
(b) Automaton $\mathcal{A}_0$



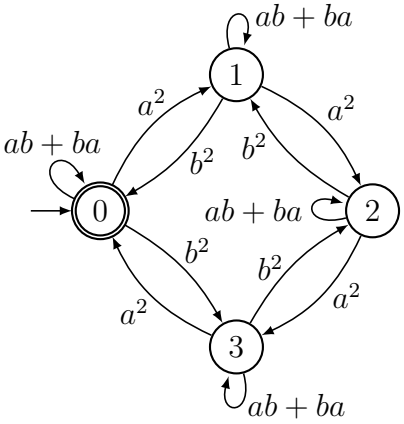
(c)



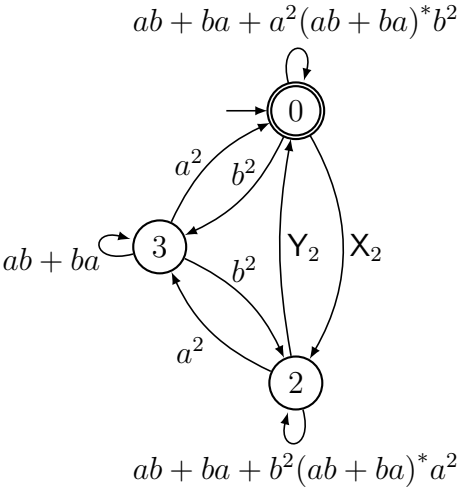
(d)



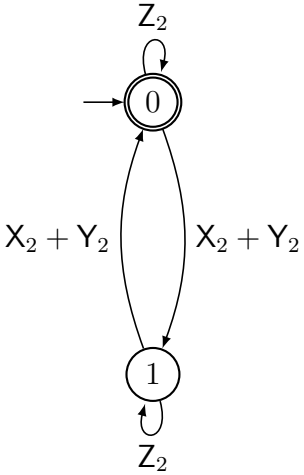
(e)



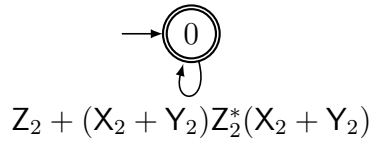
(f) Automaton $\mathcal{A}_1$



(g)



(h) Automaton $\mathcal{A}_2$



(i)

Figure 2.1: Steps of the state removal algorithm on $\mathcal{R}(8)$

2.3 Witness words

We have found a rational expression of star height q denoting the language W_q, which means that the star height of the language will not be higher then q. In the following sections we proceed to show that it also has to be at least q. First, to simplify the proof, we define a special words, show that they are in every language W_q, and show some properties of factors of those words.

Definition. For any non-negative integers k and n, *witness word* $w_{k,n}$ is a word defined recursively as:

$$\begin{aligned} w_{0,n} &= ab, \\ w_{1,n} &= a^2(ab)^n b^2 (ab)^n, \\ &\vdots \\ w_{k,n} &= a^{2^k} (w_{k-1,n})^n b^{2^k} (w_{k-1,n})^n. \end{aligned}$$

□

Lemma 9. *For any non-negative integers k and n, $|w_{k,n}|_a = |w_{k,n}|_b$.*

Proof. By induction on k. For $k = 0$, $|ab|_a - |ab|_b = 0$ regardless of n. Given the induction hypothesis for some $k-1$, $|w_{k-1,n}|_a - |w_{k-1,n}|_b = 0$. The word $w_{k,n}$ consists wholly of factors a^{2^k}, b^{2^k}, and two $(w_{k-1,n})^n$. Thus:

$$\begin{aligned} |w_{k,n}|_a - |w_{k,n}|_b &= |a^{2^k}|_a + 2|(w_{k-1,n})^n|_a + |b^{2^k}|_a - |a^{2^k}|_b - 2|(w_{k-1,n})^n|_b - |b^{2^k}|_b \\ &= 2^k + 2\,n\,|w_{k-1,n}|_a - 2\,n\,|w_{k-1,n}|_b - 2^k \\ &= 2\,n\,(\,|w_{k-1,n}|_a - |w_{k-1,n}|_b\,) = 0. \end{aligned}$$

■

From Lemma 9 it follows that any witness word $w_{k,n}$ is in any language W_q.

Lemma 10. *Any prefix u and suffix v of $(w_{k,n})^n$ satisfy the inequalities:*

$$0 \le |u|_a - |u|_b \le 2^{k+1} - 1\,, \qquad 0 \le |v|_b - |v|_a \le 2^{k+1} - 1.$$

Proof. First, we prove the inequalities of the prefix u. Let $u = (w_{k,n})^i u'$ such that u' is a proper prefix of $w_{k,n}$. Since

$$\begin{aligned} |u|_a - |u|_b &= |(w_{k,n})^i|_a + |u'|_a - |(w_{k,n})^i|_b - |u'|_b \\ &= i\,(\,|w_{k,n}|_a - |w_{k,n}|_b\,) + |u'|_a - |u'|_b \\ &= |u'|_a - |u'|_b, \end{aligned}$$

we only need to prove that the inequalities hold for a prefix of $w_{k,n}$. This lets us assume, without loss of generality, that u is proper prefix of $w_{k,n}$. We proceed by induction on k. For $k = 0$, u is a prefix of ab and we can see that the assertion is true. Given the induction hypothesis for some $k-1$, a prefix of $w_{k-1,n}$ has between 0 and $2^k - 1$ more a's than b's. There are four possibilities:

(i) u is a prefix of a^{2^k},

(ii) a^{2^k} is a prefix of u and u is a prefix of $a^{2^k}(w_{k-1,n})^n$,

(iii) $a^{2^k}(w_{k-1,n})^n$ is a prefix of u and u is a prefix of $a^{2^k}(w_{k-1,n})^n b^{2^k}$,

(iv) $a^{2^k}(w_{k-1,n})^n b^{2^k}$ is a prefix of u and u is a prefix of $a^{2^k}(w_{k-1,n})^n b^{2^k}(w_{k-1,n})^n$.

For (i), let $u = a^i$, for $0 \leq i \leq 2^k$. Then $0 \leq |u|_a - |u|_b = |u|_a \leq 2^k < 2^{k+1} - 1$.

For (ii), let $u = a^{2^k}(w_{k-1,n})^i g$, for $0 \leq i \leq n$, such that word g is a prefix of $w_{k-1,n}$. Then $0 < 2^k \leq |u|_a - |u|_b \leq 2^k + 2^k - 1 = 2^{k+1} - 1$.

For (iii), let $u = a^{2^k}(w_{k-1,n})^n b^i$, for $0 \leq i \leq 2^k$. Then

$$\begin{aligned} |u|_a - |u|_b &= |a^{2^k}|_a + |(w_{k-1,n})^n|_a + |b^i|_a - |a^{2^k}|_b - |(w_{k-1,n})^n|_b - |b^i|_b \\ &= 2^k - i\,. \end{aligned}$$

Therefore $0 \leq |u|_a - |u|_b \leq 2^k < 2^{k+1} - 1$.

For (iv), let $u = a^{2^k}(w_{k-1,n})^n b^{2^k}(w_{k-1,n})^i g$, for $0 \leq i \leq n$, such that word g is a prefix of $w_{k-1,n}$. Then

$$\begin{aligned} |u|_a - |u|_b = \;& |a^{2^k}|_a + |(w_{k-1,n})^n|_a + |b^{2^k}|_a + |(w_{k-1,n})^i|_a + |g|_a \\ & -|a^{2^k}|_b - |(w_{k-1,n})^n|_b - |b^{2^k}|_b - |(w_{k-1,n})^i|_b - |g|_b \\ = \;& 2^k + |g|_a + (n+i)(|w_{k-1,n}|_a - |w_{k-1,n}|_b) - 2^k - |g|_b \\ = \;& |g|_a - |g|_b\,. \end{aligned}$$

Therefore $0 \leq |u|_a - |u|_b \leq 2^k - 1 < 2^{k+1} - 1$.

For the proof of inequalities of suffix v, we write $w_{k,n}$ as $w_{k,n} = gv$. Since $0 = |w_{k,n}|_b - |w_{k,n}|_a = |g|_b + |v|_b - |g|_a - |v|_a$,

$$|v|_b - |v|_a = |g|_a - |g|_b.$$

Word g is a prefix of $w_{k,n}$, therefore $0 \leq |v|_b - |v|_a \leq 2^{k+1} - 1$. ■

2.4 Infinite hierarchy

We use the witness words from the previous section to define properties of languages. Each of those properties specifies a family of languages with a common star height that are all subsets of W_q. We show that for the language W_q, there are q different families, each with the common star height strictly higher then the previous one, proving that W_q itself has to have star height of at least q.

Definition. We say that language L *satisfies property* P_k if there exists an infinite number of values of n such that $(w_{k,n})^n$ is a factor of at least one word in L. □

Note that if L satisfies P_k, it also satisfies P_l for $l < k$ since $(w_{l,n})^n$ is a factor of $(w_{k,n})^n$.

Proof of Theorem 6. Lemma 8 shows that the star height of W_q is at most q. Now we show that it also has to be at least q.

By $\mathfrak{W}_k$ we denote a family of languages L that satisfy the following conditions:

(i) $L \subseteq W_q$,

(ii) L satisfies P_k,

(iii) L has a minimum star height of any language satisfying the first two conditions.

Let h_k be the common value of the star height of the languages in $\mathfrak{W}_k$. Languages in $\mathfrak{W}_0$ are necessarily infinite, therefore $0 < h_0$. P_k implies P_{k-1}, which means that $h_{k-1} \leq h_k$. W_q obviously satisfies (i). It also satisfies P_{q-1} since $|(w_{q-1,n})^n|_a = |(w_{q-1,n})^n|_b$ for every n. Because we have found a rational expression of star height q denoting W_q, it follows that $h_{q-1} \leq q$. Therefore we have

$$0 < h_0 \leq h_1 \leq \cdots \leq h_{q-1} \leq q.$$

To prove the theorem it is enough to show that $h_{k-1} < h_k$ for each k, $k = 1, \ldots, q-1$. Let L be in $\mathfrak{W}_k$. Due to Lemma 2 L can be written as a finite union of languages E_j, each denoted by a rational expression of the form

$$\mathsf{E}_j = u_0(\mathsf{H}_1)^* u_1 \cdots u_{m-1}(\mathsf{H}_m)^* u_m,$$

where each rational expression H_i denotes a language H_i with star height less then or equal to $h_k - 1$. Since L satisfies P_k and the union of the languages E_j is finite, it follows that at least one of E_j has to satisfy P_k. We can therefore safely assume that L itself is denoted by a rational expression of the same form as E_j.

For each word g in H_i, words $u_0u_1 \cdots u_m$ and $u_0u_1 \cdots u_{i-1}gu_i \cdots u_m$ are both in language L. Therefore, because it has to be true that $|g|_a \equiv |g|_b \pmod{2^q}$, $H_i \subseteq W_q$.

Since L is of a minimal star height to satisfy P_k, none of the H_i satisfies P_k. Now we need to show that some H_i satisfies P_{k-1}. In fact, we will have $h_{k-1} \leq h_k - 1$.

L satisfies P_k, therefore words $(w_{k,n})^n$ are factors of L for arbitrarily large n. Lemma 3 shows that we can find N large enough, such that there is infinitely many $n' \geq N$, that $(w_{k,n'})^{n'}$ is a factor of L, and, for each n', we have infinitely many l's, that $(w_{k,n'})^l$ is a factor of L.

Let H_i^* be the language recognised by H_i^*. Since m is finite and fixed for L, there has to be infinitely many n's, that $(w_{k,n})^n$ is a factor of $r_n \in H_i^*$ for some i, $1 \leq i \leq m$. This means that H_i^* satisfies P_k. Next we show that for these n's, $(w_{k-1,n})^n$ is a factor of a word in H_i, meaning H_i satisfies P_{k-1}.

We write r_n as a factorisation $r_n = g_0g_1 \cdots g_l$, where $g_j \in H_i$, for $0 \leq j \leq l$. If $w_{k,n}$, from $(w_{k,n})^n$, is a factor of some g_j, the condition is satisfied, since $(w_{k-1,n})^n$ is a factor of $w_{k,n}$. Otherwise, let us show how a factor $(w_{k,n})^2$ is covered by the factorisation of r_n. Written explicitly, we have

$$(w_{k,n})^2 = a^{2^k}(w_{k-1,n})^n b^{2^k}(w_{k-1,n})^n a^{2^k}(w_{k-1,n})^n b^{2^k}(w_{k-1,n})^n.$$

Let us consider the g_j that covers, at least partially, the factor b^{2^k}. There are two possibilities:

(i) b^{2^k} is a factor of g_j,

(ii) a left factor of b^{2^k} is a right factor of g_j.

In case (i), we have $g_j = vb^{2^k}u$. If v covers the factor $(w_{k-1,n})^n$ to the left of b^{2^k}, or u covers the factor $(w_{k-1,n})^n$ to the right of b^{2^k}, the condition is satisfied. If not, u is a left factor of $(w_{k-1,n})^n$ and v is a right factor. Set

$$x = |g_j|_b - |g_j|_a, \quad y = |u|_a - |u|_b, \quad \text{and} \quad z = |v|_b - |v|_a.$$

Hence

$$x = 2^k - (y - z).$$

We see that $1 - 2^k \leq y - z \leq 2^k - 1$, from Lemma 10. That gives us

$$0 < x < 2^{k+1} \leq 2^q.$$

which contradicts the fact that $g_j \in H_i \subseteq W_q$.

In case (ii), we have $g_j = vb^r$, $0 < r < 2^k$. If $(w_{k-1,n})^n$ is a factor of v, the condition is satisfied. Otherwise v is a right factor of $(w_{k-1,n})^n$ and, similarly as above, we set

$$x = |g_j|_b - |g_j|_a, \quad \text{and} \quad z = |v|_b - |v|_a.$$

Hence

$$x = r + z.$$

Therefore, due to Lemma 10,

$$r \leq x \leq r + 2^k - 1 < 2^{k+1} \leq 2^q.$$

which produces the same contradiction as the case (i). ■

3. Generalisation

The definition of star height can be expanded into *generalised star height*, a concept that is quite similar, yet current understanding of its properties is much more limited.

3.1 Generalised star height

Definition. *A generalised rational expression over* A is a formula obtained inductively from the letters of A and the symbols $\{\,\mathsf{0}\,,\,\mathsf{1}\,,+\,,\cdot\,,*\,,-\,,\wedge\,,'\,\}$ by the following process:

(i) $\mathsf{0}, \mathsf{1}$, and a, for a in A, are generalised rational expressions,

(ii) if E and F are generalised rational expressions, then

$$(\mathsf{E}+\mathsf{F})\,,\quad (\mathsf{E}\cdot\mathsf{F})\,,\quad (\mathsf{E}^*)\,,\quad (\mathsf{E}-\mathsf{F})\,,\quad (\mathsf{E}\wedge\mathsf{F})\,,\quad (\mathsf{E}')$$

are generalised rational expressions.

We write $\mathsf{GRatE}A^*$ for the set of generalised rational expressions over A. □

The languages denoted by generalised rational expressions are the same as for the rational expressions plus, to accommodate the added symbols, we let:

$$L[\mathsf{E}-\mathsf{F}] = L[\mathsf{E}] \setminus L[\mathsf{F}]\,,\quad L[\mathsf{E}\wedge\mathsf{F}] = L[\mathsf{E}]\cap L[\mathsf{E}]\,,\quad L[\mathsf{E}'] = \complement_{A^*}L[\mathsf{E}].$$

Definition. Let $\mathsf{E}\in\mathsf{GRatE}A^*$. *Generalised star height*, written $\mathsf{gsh}[\mathsf{E}]$, is defined by induction on the complexity of expressions:

$$\begin{array}{ll}
\text{if } \mathsf{E}=\mathsf{0},\ \mathsf{E}=\mathsf{1}\ \text{ or }\ \mathsf{E}=a,\ \text{for } a\in A\,, & \mathsf{gsh}[\mathsf{E}]=0\,,\\
\text{if } \mathsf{E}=\mathsf{F}+\mathsf{G}\ \text{ or }\ \mathsf{E}=\mathsf{F}\cdot\mathsf{G}\,, & \mathsf{gsh}[\mathsf{E}]=\max\left(\mathsf{gsh}[\mathsf{F}],\mathsf{gsh}[\mathsf{G}]\right)\,,\\
\text{if } \mathsf{E}=\mathsf{F}^*\,, & \mathsf{gsh}[\mathsf{E}]=1+\mathsf{gsh}[\mathsf{F}]\,,\\
\text{if } \mathsf{F}=\mathsf{E}'\,, & \mathsf{gsh}[\mathsf{E}]=\mathsf{gsh}[\mathsf{F}]\,.
\end{array}$$

□

For $\mathsf{E}=\mathsf{F}-\mathsf{G}$, or $\mathsf{E}=\mathsf{F}\wedge\mathsf{G}$, due to De Morgan's laws, identities exist:

$$\mathsf{F}-\mathsf{G}\equiv\mathsf{F}\wedge\mathsf{G}'\,,\quad\text{and}\quad \mathsf{F}\wedge\mathsf{G}\equiv(\mathsf{F}'+\mathsf{G}')'\,,$$

thus $\mathsf{gsh}[\mathsf{E}] = \max\left(\mathsf{gsh}[\mathsf{F}],\mathsf{gsh}[\mathsf{G}]\right)$. The generalised star height of rational language L over A^*, written $\mathsf{gsh}[L]$, is the minimum of the generalised star heights of the generalised rational expressions that denote L:

$$\mathsf{gsh}[L] = \min\{\mathsf{gsh}[\mathsf{E}]\mid \mathsf{E}\in\mathsf{GRatE}A^* : L = L[\mathsf{E}]\}\,.$$

3.2 Open questions

The determination of the star height of a rational language, however a difficult problem, has been solved. The determination of the generalised star height still remains to be solved. Two open questions, that are raised by the definition of generalised star height of rational language, are namely the existence of an infinite hierarchy and the computation of generalised star height.

Not only do we not know whether there are rational languages with arbitrarily large generalised star height, but even language with generalised star height greater than 1 has not yet been shown. On the other hand, for languages whose generalised star height is equal to 0, or so called *star-free languages*, Schützenberger [10] provided an algebraic characterisation.

Conclusion

This thesis proves, in greater detail than the original works, that there is an infinite hierarchy of star heights of languages over binary alphabet.

In the first chapter, we defined an unusual modification of classical automata with transitions labelled with rational expressions. This let us throughout the thesis consider only automata with at most one transition between each pair of states. We devised a precise definition of a *ring automaton* and proved Lemma 4 about the existence of computations in ring automata. For the state removal algorithm we compared the successful computations in n and $n-1$ state automata to prove their equivalence.

In the second chapter, the aim of this thesis was to take the proof of existence of languages with arbitrarily high star height, as formulated by Sakarovitch [2], and in detail prove the parts that were only outlined. In the Section 2.1 we used our Lemma 4 to show that language W_q is recognised by the ring automaton with 2^q states. In the Section 2.2 we showed that, with a particular order of states chosen for state removal algorithm, language W_q is denoted by a rational expression of star height q. In the Section 2.3 witness words are defined. In Lemma 9 we have shown that each of them has the same number of both letters of the binary alphabet over which they are defined. This equality is used to simplify the proof of Lemma 10, where we show how much more letters of one type can be in either prefix, or suffix of the witness words. This inequality is used in the proof of Theorem 6.

Bibliography

[1] F. Dejean and M.P. Schützenberger. On a question of Eggan. *Information and Control*, 9(1):23–25, 1966.

[2] J. Sakarovitch. *Elements of Automata Theory*. 1st Edition. Cambridge University Press, 2009.

[3] S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, pages 3–41, 1956.

[4] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Math. J.*, 10(4):385–397, 12 1963.

[5] R. McNaughton. The Loop Complexity of Pure-Group Events. *Information and Control*, 11(1):167 – 176, 1967.

[6] K. Hashiguchi. Regular languages of star height one. *Information and Control*, 53(3):199 – 210, 1982.

[7] K. Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124 – 169, 1988.

[8] D. Kirsten. Distance desert automata and the star height problem. *RAIRO - Theoretical Informatics and Applications*, 39(3):455–509, 2005.

[9] J. A. Brzozowski and E. J. McCluskey. Signal Flow Graph Techniques for Sequential Circuit State Diagrams. *IEEE Transactions on Electronic Computers*, EC-12(2):67–76, April 1963.

[10] M.P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190 – 194, 1965.