

# **Theoretical Computerscience – Summary**

WS 24/25

---

## Contents

<b>1 Words</b>	<b>4</b>
<b>2 Regular Languages</b>	<b>5</b>
<b>3 Regular Expressions</b>	<b>6</b>
<b>4 The Pumping Lemma</b>	<b>7</b>
<b>TODO: Add chapter contents</b>	<b>7</b>
<b>5 Equivalence Relations</b>	<b>8</b>
<b>TODO: Add chapter contents</b>	<b>8</b>
<b>6 Myhill Nerode</b>	<b>9</b>
<b>TODO: Add chapter contents</b>	<b>9</b>
<b>7 Limits of computability</b>	<b>10</b>
<b>TODO: Add chapter contents</b>	<b>10</b>
<b>8 FOR and WHILE programs</b>	<b>11</b>
<b>TODO: Add chapter contents</b>	<b>11</b>
<b>9 Syntactic Sugar</b>	<b>12</b>
<b>TODO: Add chapter contents</b>	<b>12</b>
<b>10 Gödel numberings</b>	<b>13</b>
<b>TODO: Add chapter contents</b>	<b>13</b>
<b>11 Diagonalisation</b>	<b>14</b>
<b>TODO: Add chapter contents</b>	<b>14</b>
<b>12 Universal WHILE Program</b>	<b>15</b>
<b>TODO: Add chapter contents</b>	<b>15</b>
<b>13 Halting Problem</b>	<b>16</b>
<b>TODO: Add chapter contents</b>	<b>16</b>
<b>14 Reductions</b>	<b>17</b>
14.1 Known problems . . . . .	17
14.1.1 Verification Problem $V$ . . . . .	17
14.1.2 Special Verification Problem $V_0$ . . . . .	17
14.1.3 Program termination $T$ . . . . .	17

<b>TODO: Add chapter contents</b>	<b>17</b>
14.2 Many-one Reduction . . . . .	17
14.3 Known Reductions . . . . .	17
<b>15 Common Proof Techniques</b>	<b>18</b>
15.1 Pumping Lemma . . . . .	18
15.1.1 Example . . . . .	18
15.2 Myhill Nerode . . . . .	18
15.2.1 Example . . . . .	18
<b>TODO: Add example for Myhill Nerode</b>	<b>18</b>
15.3 Reductions . . . . .	19
15.3.1 Example . . . . .	19
<b>16 Useful Proofs</b>	<b>20</b>
16.1 Regular Languages . . . . .	20
16.1.1 Finite Set . . . . .	20
16.1.2 Finite Automaton . . . . .	20
16.1.3 Regular Expression . . . . .	20
16.1.4 Closure Properties . . . . .	20
16.2 Non-Regular Languages . . . . .	21
16.2.1 Pumping Lemma . . . . .	21
16.2.2 Myhill Nerode . . . . .	21
<b>TODO: Add solution for Myhill Nerode</b>	<b>21</b>
<b>Index</b>	<b>22</b>

## 1 Words

A word  $w$  (also called String) has length  $l$  and consists of symbols  $\sigma \in \Sigma$ .

The empty word  $\varepsilon$  has length 0.

## **2 Regular Languages**

### 3 Regular Expressions

A regular expression always describes a regular language. If we can build a regular expression  $E$ , then  $L(E) \in \text{REG}$ .

## 4 The Pumping Lemma

**TODO: Add chapter contents**

## 5 Equivalence Relations

**TODO: Add chapter contents**



## 6 Myhill Nerode

**TODO: Add chapter contents**

## 7 Limits of computability

**TODO: Add chapter contents**

## 8 FOR and WHILE programs

**TODO: Add chapter contents**

## 9 Syntactic Sugar

**TODO: Add chapter contents**

## 10 Gödel numberings

**TODO: Add chapter contents**

## 11 Diagonalisation

**TODO: Add chapter contents**

## 12 Universal WHILE Program

**TODO: Add chapter contents**

## 13 Halting Problem

**TODO: Add chapter contents**



## 14 Reductions

### 14.1 Known problems

#### 14.1.1 Verification Problem $V$

The verification problem has two gödel number inputs namely  $i, j$  and checks if the two programs corresponding to these gödel numbers output the same for every possible input.

#### 14.1.2 Special Verification Problem $V_0$

The special verification problem has a gödel number  $i$  as input and checks if the program corresponding to that gödel number will output 0 for every possible input.

#### 14.1.3 Program termination $T$

The program termination problem has a gödel number  $i$  as input and checks if the program corresponding to that gödel number will terminate for every possible input.

**TODO: Add chapter contents**

### 14.2 Many-one Reduction

A WHILE computable total function  $f : \mathbb{N} \rightarrow \mathbb{N}$  is called many-one reduction from  $L$  to  $L'$  if

- $L, L' \subseteq \mathbb{N}$
- $\forall x \in \mathbb{N} : x \in L \iff f(x) \in L'$

If such an  $f$  exists, then  $L$  is many-one reducible to  $L'$ , therefore we can write  $L \leq L'$

### 14.3 Known Reductions

$$H_0 \leq V_0$$

$$V_0 \leq V$$

$$V_0 \leq T$$

$$\overline{H_0} \leq V_0$$

## 15 Common Proof Techniques

### 15.1 Pumping Lemma

The pumping lemma can only be used to show that a language is not REG. We take a word  $w \in L$  and split it into multiple sub words  $u, v, w$  where  $|v| \geq n$  with  $n \geq 0$ . Now there are words  $x, y, z$  with  $v = xyz$  and  $|y| \geq 0$  such that  $uxy^i zw \in L$  for all  $i \in \mathbb{N}$ . Afterwards we try to find an  $i$  such that the pumped word is no longer  $\in L$  and thus we proved that the language is not regular.

#### 15.1.1 Example

##### Exercise:

Show that the following language is not regular.

Let  $A = \{1^{(3n)} \mid n \in \mathbb{N}\}$

##### Solution:

Let  $n \geq 0$  be given.

We choose  $u = 1^n, v = 1^n, w = 1^n$  such that  $uvw = 1^{3n}$  and  $|v| = n$ .

Let  $x, y, z$  be given as  $x = 1^r, y = 1^s, z = 1^t$  with  $xyz = v$  and  $s \geq 0$  since  $y \neq \epsilon$  and  $r + s + t = n$ .

$$uxy^i zw = 1^n 1^r 1^{s \cdot i} 1^t 1^n = 1^n 1^{r+s+t} 1^{s \cdot (i-1)} 1^n$$

We choose  $i = 0$ , therefore  $1^n 1^{r+s+t} 1^{s \cdot (i-1)} 1^n = 1^n 1^{n-s} 1^n \notin A$  since it is not of the form  $1^{3m}$  anymore for any  $m \in \mathbb{N}$ .

Therefore we cannot pump language  $A$  and thus it is not regular.

### 15.2 Myhill Nerode

#### 15.2.1 Example

TODO: Add example for Myhill Nerode

## 15.3 Reductions

### 15.3.1 Example

**Exercise:**

Consider the following language  $L = \{i \in \mathbb{N} \mid \text{göd}^{-1}(i) \text{ outputs 42 on input 1337}\}$

Prove  $L \notin \text{co-RE}$

**General Ideas:**

$\bar{L}$  is the set of all programs that either diverge or output something else than 42 on input 1337.

$L$  is the set of all programs that output 42 on input 1337 which is equal to running a sub program that halts on every input and outputting 42 afterwards. Thus we could try to reduce the special halting problem  $H_0$  to  $L$ .

**Solution:**

$$\forall L : L \in \text{REC} \iff L \in \text{RE} \wedge \bar{L} \in \text{RE}$$

$$L \notin \text{co-RE} \iff \bar{L} \in \text{RE}$$

$H_0 \notin \text{REC}$  but  $H_0 \in \text{RE}$ , therefore  $\overline{H_0} \notin \text{RE}$  thus reducing  $\overline{H_0} \leq \bar{L}$  suffices to show that  $\bar{L} \notin \text{RE}$  but this is the same as  $H_0 \leq L$  by equivalence.

The reduction is given by  $f(i) := \text{göd}(P)$

We create the WHILE program  $P$ , that has input  $m$ . However we ignore the input  $m$  and simulate  $i$  on the input  $i$ , then return 42. Thus  $f$  is obviously WHILE computable.

We consider the following two cases:

- Let  $i \in H_0$ , then  $i$  halts on input  $i$  by definition. Program  $P$  halts on arbitrary inputs  $m$  and outputs 42. Therefore  $P$  also outputs 42 on our input 1337.  
Thus  $\text{göd}(P) \in L$  holds.
- Let  $i \notin H_0$ , then  $i$  does not halt on input  $i$  by definition. Therefore Program  $P$  diverges on all inputs, which is also the case for input 1337.  
Therefore  $P$  diverges on our input 1337.  
Thus  $\text{göd}(P) \notin L$  holds.

Therefore we have successfully shown that  $H_0 \leq L$  is a valid reduction.

## 16 Useful Proofs

### 16.1 Regular Languages

#### 16.1.1 Finite Set

**Exercise:**

Show that the following language is regular over the alphabet  $\{0,1\}$ .

$$L = \{x \mid x \text{ is prime and } x < 1'000'000'000\}$$

**Solution:**

Since there are only finitely many prime numbers between 0 and  $1'000'000'000$ , the set of the words that are accepted by  $L$  is finite and thus the language is regular.

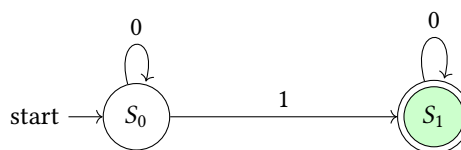
#### 16.1.2 Finite Automaton

**Exercise:**

Show that the following language is regular over the alphabet  $\{0,1\}$ .

$$L = \{0^n 10^m \mid n, m \in \mathbb{N}\}$$

**Solution:**



Since we can describe the language  $L$  by the finite automaton given above, the language is regular.

#### 16.1.3 Regular Expression

**Exercise:**

Show that the following language is regular over the alphabet  $\{0,1\}$ .

$$L = \{0^n 1^m \mid n, m \in \mathbb{N}\}$$

**Solution:**

Let  $E = 0^*1^*$  be the regular expression describing the language  $L$ .

Since we can describe the language  $L$  by the regular expression given above, the language is regular.

#### 16.1.4 Closure Properties

## 16.2 Non-Regular Languages

### 16.2.1 Pumping Lemma

**Exercise:**

Show that the following language is not regular over the alphabet  $\{0,1\}$ .

$$L = \{0^n 1^n \mid n \in \mathbb{N}\}$$

**Solution:**

Let  $n \geq 0$  be given.

We choose  $u = 1^n$ ,  $v = 1^n$ ,  $w = 1^n$  such that  $uvw = 1^{3n}$  and  $|v| = n$ .

Let  $x,y,z$  be given as  $x = 1^r$ ,  $y = 1^s$ ,  $z = 1^t$  with  $xyz = v$  and  $s \geq 0$  since  $y \neq \varepsilon$  and  $r + s + t = n$ .

$$uxy^i z w = 1^n 1^r 1^{s \cdot i} 1^t 1^n = 1^n 1^{r+s+t} 1^{s \cdot (i-1)} 1^n$$

We choose  $i = 0$ , therefore  $1^n 1^{r+s+t} 1^{s \cdot (i-1)} 1^n = 1^n 1^{n-s} 1^n \notin A$  since it is not of the form  $1^{3m}$  anymore for any  $m \in \mathbb{N}$ .

Therefore we cannot pump language  $A$  and thus it is not regular.

### 16.2.2 Myhill Nerode

**Exercise:**

Show that the following language is not regular over the alphabet  $\{0,1\}$ .

$$L = \{0^n 1^n \mid n \in \mathbb{N}\}$$

**Solution:**

**TODO: Add solution for Myhill Nerode**

# Index

Common Proof Techniques, 18

- Myhill Nerode, 18

- Example, 18

- Pumping Lemma, 18

- Example, 18

- Reductions, 19

- Example, 19

Diagonalisation, 14

Equivalence Relations, 8

FOR and WHILE programs, 11

Gödel numberings, 13

Halting Problem, 16

Limits of computability, 10

Myhill Nerode, 9

Reductions, 17

- Known problems, 17

- Program termination  $T$ , 17

- Special Verification Problem  $V_0$ , 17

- Verification Problem  $V$ , 17

- Known Reductions, 17

- Many-one Reduction, 17

Regular Expressions, 6

Regular Languages, 5

Syntactic Sugar, 12

The Pumping Lemma, 7

Universal WHILE Program, 15

Useful Proofs, 20

- Non-Regular Languages, 21

- Regular Languages, 20

- Closure Properties, 20

- Finite Automaton, 20

- Finite Set, 20

- Regular Expression, 20

Words, 4