II.4.1. Regular Grammars and NFAs (13.5)

We will show that regular expressions coincide with regular languages and with languages recognised by a DFA or NFA. Here we prove one part of this result:

**Theorem (II.4.1.1.)**

For every right linear grammar $G$ there exists an NFA $A$ s.t.

$$L(G) = L(A)$$

A can be computed from $G$. 
**Proof Idea**

- A derivation of a word in $G$ has the form
  \[ S = A_0 \rightarrow a_1 A_1 \rightarrow a_1 a_2 A_2 \rightarrow \cdots \rightarrow a_1 a_2 \cdots a_{n-1} A_{n-1} \rightarrow a_1 a_2 \cdots a_n \]
  where we have productions
  \[ A_i \rightarrow a_{i+1} A_{i+1} \quad A_{n-1} \rightarrow a_n \]
  or
  \[ S = A_0 \rightarrow a_1 A_1 \rightarrow a_1 a_2 A_2 \rightarrow \cdots \rightarrow a_1 a_2 \cdots a_{n-1} A_{n-1} \rightarrow a_1 a_2 \cdots a_{n-1} \]
  where we have productions
  \[ A_i \rightarrow a_{i+1} A_{i+1} \quad A_{n-1} \rightarrow \epsilon \]

So we have:
- If $B \rightarrow aB'$, then $B \xrightarrow{a} B'$.
- If $B \rightarrow a$, then $B \xrightarrow{a} q_F$.
- $q_F \in F$.
- If $B \rightarrow \epsilon$, then $B \in F$.

**Proof Idea**

Define $A$ with states $N \cup \{q_F\}$ for a special new accepting state $q_F$ s.t. the derivation

\[ S = A_0 \rightarrow a_1 A_1 \rightarrow a_1 a_2 A_2 \rightarrow \cdots \rightarrow a_1 a_2 \cdots a_{n-1} A_{n-1} \rightarrow a_1 a_2 \cdots a_n \]

corresponds to a sequence of transitions

\[ S = A_0 \xrightarrow{a_1} A_1 \xrightarrow{a_2} A_2 \xrightarrow{a_3} \cdots \xrightarrow{a_{n-1}} A_{n-1} \xrightarrow{a_n} q_F \]

and a derivation

\[ S = A_0 \rightarrow a_1 A_1 \rightarrow a_1 a_2 A_2 \rightarrow \cdots \rightarrow a_1 a_2 \cdots a_{n-1} A_{n-1} \rightarrow a_1 a_2 \cdots a_n \]

corresponds to a sequence of transitions

\[ S = A_0 \xrightarrow{a_1} A_1 \xrightarrow{a_2} A_2 \xrightarrow{a_3} \cdots \xrightarrow{a_{n-1}} A_{n-1} \in F \]

**Constructed NFA**

We obtain from $G = (N, T, S, P)$ the following NFA:

- **automaton**: $A$
- **states**: $N \cup \{q_F\}$
- **.terminals**: $T$
- **start**: $S$
- **final**: $B \in N$ s.t. $B \rightarrow \epsilon$
- **transitions**: $B \xrightarrow{a} B'$ if $B \rightarrow aB'$.
  $B \xrightarrow{a} q_F$ if $B \rightarrow a$. 
Proof of Theorem II.4.1.1.

Can be found in the additional material.

Example

Consider the Grammar:

```
grammar G

terminals 0, 1

nonterminals S, T

start symbol S

productions S → 0, S → 1T, 
T → 0T, T → 1T,
T → ε, T → 0, T → 1
```

Corresponding Automaton

(Note that it is nondeterministic.)

```
S 0,1
   ↓ 1
    ↓ 0
    ↓ 0,1
    ↓ 1
    ↓ 0
    ↓ 0,1
    ↓ 0
    ↓ 0
T
```

Corresponding Automaton

With corresponding rules:

```
S → 0
S → 1T
T → 0T
T → 1T
T → ε
```

Accepting state because of

```
T → ε
```
II.4.2. Translating NFAs into Regular Expressions (13.10)

**Theorem II.4.2.1.**

Let \( A = (Q, q_0, F, T, \rightarrow) \) be an NFA. Then there exists a regular expression \( E \) s.t. \( L(E) = L(A) \). \( E \) can be computed from \( A \).

**Proof of Theorem II.4.2.1.**

A proof of Theorem II.4.2.1. and an example can be found in the additional material.
Theorem II.4.3.1.

Theorem (II.4.3.1.)

Let $L$ be a language over an alphabet $T$. The following are equivalent:

1. $L$ is definable by a regular expression.
2. $L$ is a regular.
3. $L$ is definable by a right-linear grammar.
4. $L$ is definable by a left-linear grammar.
5. $L$ is definable by an NFA with empty moves.
6. $L$ is definable by an NFA.
7. $L$ is definable by a DFA.

Furthermore, the corresponding regular expressions, right linear grammars, left-linear grammars, NFAs with empty moves, NFAs, DFAs can be computed from each other.

Proof of Theorem II.4.3.1.

A proof of Theorem II.4.3.1. can be found in the additional material.

II.4.4. Closure Properties and Decidability of Regular Languages

Closure Properties

Theorem (II.4.4.1.)

Regular languages are closed under

1. complement,
2. intersection,
3. the operation $L \mapsto L^R$.

So if $L, L'$ are regular languages over alphabet $T$, so are

1. $L^c$ (the complement of $L$, i.e. $\{ t \in T^* | t \not\in L \}$),
2. $L \cap L'$,
3. $L^R$ (i.e. $\{ w^R | w \in L \}$, where $w^R$ is the result of reverting $w$).

Furthermore, regular grammars for $L^c$, $L \cap L'$ and $L^R$ can be computed from those for $L$. 
Proofidea for Theorem II.4.4.1.

- We will use that regular expressions, languages definable by DFAs and regular languages are equivalent.
- Languages definable by regular expressions are closed under $L \mapsto L^R$.
  (See additional material, Lemma II.4.3.4).
- One can easily see that languages defineable by a DFA are closed under $L \mapsto L^c$ and under intersection.
- Therefore the assertion follows.
- Full details can be found in the additional material.

Decision Problems

Theorem (II.4.4.3.)

- We can decide for regular languages whether $L = \emptyset$.
- We can decide for regular languages $L$ and $L'$ whether $L \subseteq L'$.
- We can decide for regular languages $L$ and $L'$ whether $L = L'$.

Proof of Theorem II.4.4.3.

A proof of Theorem II.4.4.3. can be found in the additional material.
II.4.5. The Pumping Lemma for Regular Languages (12.4, 12.5)

Motivation

- We want to show that there are languages which are context-free but not regular.
- In order to do this we prove the pumping lemma, which uses the fact that an NFA has only finitely many states.
  (We could use as well the fact that a regular grammar has only finitely many nonterminals).

- **Note** The following slides contain some coloured parts. The colours are indistinguishable in the black and white handouts. It is recommended to look at them using the online version.

- Reminder on notations:
  - $|u|$ is the length of string $u$.
  - $uv$ denotes the concatenation of strings $u$ and $v$.

II.4.5. The Pumping Lemma for Regular Languages (12.4, 12.5)

Using the Finiteness of an NFA

Consider an NFA

This NFA has 5 states.
Any run of the NFA for a word of length $\geq 5$ uses at least 6 states. Therefore it must visit one state at least twice. So there must be a loop within the first 5 letters of such a word.

Here is the run for the word $z = \text{ababa}$ using colours blue, red and green.

- The **blue part** is the part before we reached a state visited twice, corresponding to the word $u = a$.
- The **red part** is the part from the state visited twice until we reach it again, corresponding to the word $v = bab$.
- The **green part** is the remaining part, corresponding to the word $w = a$.
- The loop must occur within the first 5 letters, so $|uv| \leq 5$. Because $v$ is along a loop, $|v| \geq 1$.

- If we repeat the loop several times, we obtain as well an accepting word.
  - If we start with $u = a$, then repeat the loop following the word $v = bab$ $i$ times, then follow the word $w = a$, we obtain an accepting run.
  - It accepts the word $a(bab)^i a$.
    - E.g. in case $i = 2$ the word is $abababa$.
    - In case $i = 0$ the word is $aa$.
  - In general we get that the word $uv'w$ is an element of the language as well.

CS_275 Sect. (II.4.5.) 25/34

CS_275 Sect. (II.4.5.) 26/34

CS_275 Sect. (II.4.5.) 27/34

CS_275 Sect. (II.4.5.) 28/34
II.4.5. The Pumping Lemma for Regular Languages (12.4, 12.5)

Generalisation

Assume an NFA $A$ having $k$ states. Then for every word $x \in L(A)$ s.t. $|x| \geq k$ there exist words $u, v, w$ s.t.

$$x = uvw, \quad |uv| \leq k, \quad |v| \geq 1$$

and s.t.

$$uv^iw \in L(A) \text{ for all } i \in \mathbb{N}$$

This follows by the above considerations.

So we have proved the following theorem:

**Theorem (Pumping Lemma for Regular Languages)**

Let $L$ be a regular language. Then there exist a fixed number $k$ depending on $L$ only s.t. we have the following:

- If $x \in L$ is a word, $|x| \geq k$, then there exist words $u, v, w$ s.t.

  $$x = uvw, \quad |uv| \leq k, \quad |v| \geq 1$$

  and s.t.

  $$uv^i w \in L(A) \text{ for all } i \in \mathbb{N}$$

Example 1

**Lemma**

The language $L := \{a^i b^i \mid i \geq 1\}$ is context-free but not regular.

**Proof (Example 1)**

- We have already seen that $L$ is context-free.
- Assume $L$ is regular.
- Let $k$ be as in the pumping lemma.
- Consider $x := a^k b^k \in L$.
  - $|x| \geq k$, so there exist $u, v, w$ s.t.
    $$x = uvw, \quad |uv| \leq k, \quad |v| \geq 1,$$
    and s.t.
    $$uv^i w \in L \text{ for all } i \in \mathbb{N}.$$  
  - Since $|uv| \leq k$, $u$ and $v$ are substrings of $a^k$.
  - Therefore $uv^2 w = a^{k+i} b^k$ where $l = |v|$.
  - But $a^{k+i} b^k \notin L$, a contradiction.
Example 2

Lemma

The language $L := \{xx^R \mid x \in \{a, b\}^*\}$ is context-free but not regular.

Proof (Example 2)

- We have already seen that $L$ is context-free.
- Assume $L$ is regular.
- Let $k$ be as in the pumping lemma.
- Consider $x := a^kbba^k \in L$.
- $|x| \geq k$, so there exist $u, v, w$ s.t. $x = uvw$, $|uv| \leq k$, $|v| \geq 1$, and s.t. $uv^i w \in L$ for all $i \in \mathbb{N}$.
- Since $|uv| \leq k$, $u$ and $v$ are substrings of $a^k$.
- Therefore $uv^2w = a^{k+l}bba^k$ where $l = |v|$.
- But $a^{k+l}bba^k \notin L$, a contradiction.