Final lecture: Formal Languages, Grammars and

Automata
• Pushdown Automata and Context Free Grammars (recall)
• Where does this topic go from here . . . ?
• Typical exam exercises: see the exercises of the “werkcollege”; we provide some additional ideas of exercises.
A pushdown automaton is a sextuple \( \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle \) with

- \( Q \) a finite set of states
- \( q_0 \) an element of \( Q \), the initial state
- \( F \) a subset of \( Q \)
- \( \Sigma \) a finite set of symbols (input alphabet)
- \( \Gamma \) the stack alphabet
- \( \delta \) a map ('afbeelding')

\[
\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q \times \Gamma \cup \{\lambda\})
\]

We write e.g. \( \delta(q_i, a, A) = \{[q_j, B], [q_k, C]\} \)

We understand the \( Q, \Sigma, P, \lambda \). New is the \( \Gamma \): alphabet of stack symbols

The stack is not mentioned, but it is used in the operation of the PDA!
Action of a PDA

\[ [q', B] \in \delta(q, a, A): \text{you can read } a, \text{ pop } A \text{ and push } B \]

\[ L(M) = \{ w \in \Sigma^* \mid [q_0, w, \lambda] \vdash^* [q_i, \lambda, \lambda] \& q_i \in F \} \]
• Study of languages. Natural languages and computer languages.
  When are two languages the same? How do you parse or compile a language? In what complexity class is a language?
• Extending the stack to a tape that one can navigate over: Turing Machine, universal model of computation.
  See the course IBC003: Berekenbaarheid.
• Automata and grammars as small computing devices: simple, but well-understood. Slogan:
  The basic building blocks are simple;
  Complexity is an emerging property.
  E.g. genes and phenotype, Conway’s game of life, Lindenmayer-systems (see next hour).
Typical exercises 1: Transformations (regular)

Let $L = \{ w \in \{a, b\}^* \mid \#_a(w) \text{ is even} \}$

(i) Construct a regular expression $e$ such that $L(e) = L$

(ii) Construct a DFA $M$ such that $L(M) = L$

(iii) Construct a regular grammar $G$ such that $L(G) = L$
Typical exercises 2: Transformations (context-free)

Let \( L = \{ a^n b^m \mid n > m \} \)

(i) Construct a PDA \( M \) such that \( L(M) = L \)

(ii) Construct a grammar \( G \) such that \( L(G) = L \)

(iii) Show that \( L \) is not regular.
Typical exercises 3: Equivalences

(i) Let \( e_1 = u^* (u \cup v)^* \) and \( e_2 = (u \cup vu^*)^* \), where \( u, v \in \Sigma^* \)
Show that \( L(e_1) = L(e_2) \)

(ii) Let \( M_1 = \)

and

\[ M_2 = \]
Show that \( L(M_1) = L(M_2) \)
Typical exercises 4: the ‘square’

Given is $\Sigma = \{a, b\}$

Consider

$$L = \{a^n ba^m \mid n > 0, m \geq 0\}$$

$$e = aa^* ba^*$$

$$M = \begin{array}{c}
q_0 \xrightarrow{a} a \xrightarrow{b} a_2 \xrightarrow{b} a_2
\end{array}$$

$$G = \begin{array}{c}
S \rightarrow aA \\
A \rightarrow aA \mid bB \\
B \rightarrow aB \mid \lambda
\end{array}$$

Show that $L = L(e) = L(G) = L(M)$.

[Hint. Showing $L \subseteq L(e) \subseteq L(G) \subseteq L(M) \subseteq L$ suffices!]
Typical exercises 5: Equivalences

(iii) Let $G_1 = S \rightarrow AB$ and $G_2 = S \rightarrow AB \mid A \mid B \mid \lambda$

$A \rightarrow aA \mid \lambda$  $A \rightarrow aA \mid a$

$B \rightarrow bB \mid \lambda$  $B \rightarrow bB \mid b$

Show that $L(G_1) = L(G_2)$
Typical exercises 6 (*): the 'triangle'

Given is $\Sigma = \{a, b\}$

Consider

$$L = \{ w \in \Sigma^* \mid \#_a(w) = \#_b(w) \}$$

$$M = \begin{array}{c}
S \rightarrow aA \mid bB \mid \lambda \\
A \rightarrow bS \mid aAA \\
B \rightarrow aS \mid bBB
\end{array}$$

(i) Show that $L = L(M)$

(ii) Show that $L = L(G)$