12. Exercises for the Course
‘Automata and Logic’

Exercise 57:

Example 6.10 from the lecture shows that deterministic RL-tree automata recognize a smaller class of languages than non-deterministic ones. We call an RL-tree automaton $A = (Q, \Sigma, I, \Delta, F)$ quasi-deterministic if

- $\Delta$ is a deterministic transition assignment, and
- $I \subseteq Q$ is a set of initial states.

Prove or refute:

(a) If $L \subseteq T_{\Sigma}$ is a finite tree language, then there exists a quasi-deterministic tree automaton recognizing $L$.

(b) If $L \subseteq T_{\Sigma}$ is a recognizable tree language, then there exists a quasi-deterministic tree automaton recognizing $L$.

Exercise 58:

Construct regular expressions for the following tree languages:

(a) The language of all trees that represent arithmetic expressions over the binary symbols $+$ and $\cdot$, the unary symbol $\neg$, and the variables (i.e. nullary symbols) $x$, $y$, and $z$.

(b) The language of all trees that represent regular expressions over the alphabet $\Sigma_L := \{a, b\}$.

You may add auxiliary symbols to the alphabets if necessary.

Exercise 59:

Let $\Sigma$ be an alphabet with arity function. Show the following without using Proposition 6.20:

(a) For every $t \in T_{\Sigma}$, the language $\{t\}$ is recognizable.

(b) For every $t \in T_{\Sigma}$, the language $\{t\}$ is regular.
**Exercise 60:**

Let $\Sigma = \{\lor, \land, \neg, \top, \bot\}$ be an alphabet with the obvious arity function. The following RL-tree automaton $A = (Q, \Sigma, I, \Delta, F)$ accepts precisely those trees that represent Boolean expressions evaluating to “false”:

- $Q = \{0, 1\}$,
- $I = \{0\}$,
- $\Delta_\neg(0) = \{1\}, \Delta_\neg(1) = \{0\}$
- $\Delta_\land(0) = \{(0, 0), (0, 1), (1, 0)\}, \Delta_\land(1) = \{(1, 1)\}$
- $\Delta_\lor(0) = \{(0, 0)\}, \Delta_\lor(1) = \{(0, 1), (1, 0), (1, 1)\}$
- $F(\top) = 1, F(\bot) = 0$.

Use the method from the proof of Theorem 6.20 to construct a regular expression for the tree language recognized by $A$.

**Exercise 61:**

Devise a quadratic time algorithm that decides the emptiness problem for LR-tree automata.