1 Infinite Computations

Theorem 1. *NBA-recognizable languages are closed under complement.*

Proof. □

Theorem 2.

Proof. □

Theorem 3 (Landweber)**.** *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ be a DMA.*

1. *$L(\mathcal{A})$ is DBA-recognizable iff $\mathcal{F}$ is closed under super loops.*
2. *$L(\mathcal{A})$ is E-recognizable iff $\mathcal{F}$ is closed under reachable loops.*

Proof. □

2 Tree Automata

Theorem 4. *Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ be a TWA. There is an NTA $\mathcal{A}'$ of size exponential in $|Q|$ that recognizes $T(\mathcal{A})$.*

Proof. Let $\sim_{T(\mathcal{A})}$ be the usual equivalence relation, i.e. $t_1 \sim_{T(\mathcal{A})} t_2$ iff $\forall s \in S_\Sigma : s \cdot t_1 \in T(\mathcal{A}) \leftrightarrow s \cdot t_2 \in T(\mathcal{A})$. We define a relation $\sim \subseteq T_\Sigma \times T_\Sigma$ such that $\text{index}(\sim_{T(\mathcal{A})}) \leq \text{index}(\sim) \leq 2^{|Q|^2 \cdot m+1}$, where m is the maximal rank in Σ.

Let $t_0 \in T_\Sigma$ and $a_m \in \Sigma_m$ be arbitrary. For every $t \in T_\Sigma$ and $1 \leq i \leq m$, we define $t^{(i)} = a_m(t_0, t_0, \dots, t, \dots, t_0)$, meaning the i-th subtree below the root is t. Further, we define a relation $B_t^i \subseteq Q \times Q$ with $(p, q) \in B_t^i$ iff there is a run segment ρ of $\mathcal{A}$ on $t^{(i)}$, such that the run begins at the root of t, never leaves that subtree until the end. Meaning, $\rho = (p, i)(q_1, iu_1) \dots (q_n, iu_n)(q, \varepsilon)$.

Finally, let $t_1 \sim t_2$ iff $t_1 \in T(\mathcal{A}) \leftrightarrow t_2 \in T(\mathcal{A})$ and $\forall i : B_{t_1}^i = B_{t_2}^i$.

Idea: $(p, q) \in B_t^i$ if $\mathcal{A}$ can enter t as i-th child with state p and after some while leaves it again with state q.

Claim : Let $t_1 \sim t_2$. Then $t_1 \sim_{T(\mathcal{A})} t_2$.

Let $s \in S_\Sigma$. Due to the symmetric definition of $\sim$, it suffices to show that $t_1 \in T(\mathcal{A})$ implies $t_2 \in T(\mathcal{A})$, so let $t_1 \in T(\mathcal{A})$. If $s = \circ$, then $s \cdot t_1 = t_1 \in T(\mathcal{A})$. By definition of $\sim$, this implies $s \cdot t_2 = t_2 \in T(\mathcal{A})$.

Otherwise $s \neq \circ$. Let $\rho_1\rho_2\rho_3$ be an accepting run of $\mathcal{A}$ on $s \cdot t_1$ such that ρ_1 only stays outside of t_1 and ρ_2 only stays inside of t_1. Since $B_{t_1}^i = B_{t_2}^i$, there is a run segment of $\mathcal{A}$ on t_2 which enters and exits the tree with the same states as ρ_2 does, meaning it can replace ρ_2 in the accepting run. Repeating this procedure gives an accepting run of $\mathcal{A}$ on $s \cdot t_2$, so $t_2 \in T(\mathcal{A})$.

Notes on the construction: each state in the NTA corresponds to a list of Q-states that $\mathcal{A}$ had when visiting this node, together with the direction from which it was coming. That can be used to check the correctness of a run. □

Theorem 5. *A language of finite trees $T \subseteq T_\Sigma$ can be recognized by an NTA iff it can be described by a regular tree expression.*

Proof. $\Rightarrow$ Let $\mathcal{A} = (Q, \Sigma, \Delta, F)$ be an NTA for T. For $R, I \subseteq Q, q \in Q$, we define $T(R, I, q) \subseteq T_{\Sigma \cup C_R}$ (where $C_R = \{c_p \mid p \in R\}$) as the set of all trees on which $\mathcal{A}$ has a run that only uses states in I and ends in q. We can inductively define regular expressions for $T(R, I, q)$. Then $T(\mathcal{A}) = \bigcup\limits_{q \in F} T(\emptyset, Q, q)$.

For all $R \subseteq Q, q \in Q$, $T(R, \emptyset, q)$ contains only trees of height 0 or 1, so it is a finite union of singular languages.

$$T(R, I \cup \{i\}, q) = T(R, I, q) + T(R \cup \{i\}, I, q) \cdot^{c_i} (T(R \cup \{i\}, I, i))^{*c_i} \cdot^{c_i} T(R, I, i)$$

$\Leftarrow$ Show by induction that a regular expression r can be transformed to a NTA $\mathcal{A}_r$.

- If $r = t \in T_{\Sigma\cup C}$, then there is an automaton $\mathcal{A}_t$ with $T(\mathcal{A}_t) = \{t\}$.
- If $r = s + t$, then $\mathcal{A}_r$ is the union automaton of $\mathcal{A}_s$ and $\mathcal{A}_\sqcup$.
- If $r = s \cdot^c t$, then $\mathcal{A}_r = (Q_s \cup Q_t, \Sigma \cup C, \Delta_r, F_s)$, where Δ_r simulates $\mathcal{A}_s$ and $\mathcal{A}_t$ but the final transitions in $\mathcal{A}_t$ are replaced by initial transitions to $\mathcal{A}_s$.
- If $r = s^{+_c}$, let $\mathcal{A}_r = (Q_s, \Sigma \cup C, \Delta_r, F_s)$, where Δ_r simulates $\mathcal{A}_s$ and allows "restarts" when a final state could be reached.

$\square$

Theorem 6. *Let D be an EDTD. We can construct a NUTA $\mathcal{A}$ with $T(D)^{\complement} = T(\mathcal{A})$ in polynomial time in $|D|$.*

Proof. Let $D = (\Sigma', P, S^{(1)})$. For $a^{(i)} \in \Sigma'$, let $a^{(i)} \to r_{a,i}$ be the according rule in P, where $r_{a,i}$ is a regular expression that can be transformed to a DFA in polynomial time. If $b \in \Sigma$ occurs in $r_{a,i}$ then it must have a unique type j (since D is single-typed). We call $b^{(j)}$ the b-successor of $a^{(i)}$. If b does not occur in $r_{a,i}$, we say that the b-successor is $b^{\perp}$. Furthermore, we assume $r_{a,\perp} = \varepsilon$ for all a.

We define a typing function $f : \mathrm{dom}_t \to \{\perp, 1, \ldots, k\}$. We assign $f(\varepsilon) = \begin{cases} 1 & \text{if } \mathrm{val}_t(\varepsilon) = S \\ \perp & \text{else} \end{cases}$.
For the other nodes, let u be a node with parent v. We call $\mathrm{val}_t(v) = a$ and $\mathrm{val}_t(u) = b$. Then there is a unique $i \in \{\perp, 1, \ldots, k\}$ such that $b^{(i)}$ is the b-successor of $a^{f(v)}$. We set $f(u) := i$.

Claim: $t \in T(D)$ iff $\forall v \in \mathrm{dom}_t : f(v) \neq \perp$ and $a_1^{f(v1)} \ldots a_m^{f(vm)} \in L(r_{a,f(v)})$, where $a = \mathrm{val}_t(v)$ and $a_j = \mathrm{val}_t(vj)$. (without proof)

Using this claim, we can provide an automaton $\mathcal{A} = (Q, \Sigma, \Delta, F)$ that accepts a tree iff it violates the constraint.

- $Q = \Sigma \times \{\perp, 1, \ldots, k\} \times \{0, 1\}$, where the last component denotes whether a violation was found.
- $\Delta = \{(L_{a,i,x}, a, (a, i, x)) \mid a \in \Sigma, i \in \{\perp, 1, \ldots, k\}, x \in \{0, 1\}\}$
  Let $\alpha = (a_1, i_1, x_1) \ldots (a_m, i_m, x_m) \in Q^m$ and $w := a_1^{(i_1)} \ldots a_m^{(i_m)} \in (\Sigma')^m$.
  - succ $:\Leftrightarrow$ for all $1 \leq j \leq m$, $a_j^{(i_j)}$ is the a_j-successor of $a^{(i)}$.
  - $\mathrm{sat}_0 :\Leftrightarrow w \in L(r_{a,i})$ and for all $1 \leq j \leq m$, $x_j = 0$ and $i_j \neq \perp$.
  - $\mathrm{sat}_1 :\Leftrightarrow w \notin L(r_{a,i})$ or there is a $1 \leq j \leq m$ such that $x_j = 1$ or $i_j = \perp$.

  Then $\alpha \in L_{a,i,x}$ iff succ and sat_x hold.
- $F = \{(S, 1, 1)\} \cup \{(a, i, x) \mid a \neq S\}$, meaning either a violation was found or the starting symbol was not S.

$\square$

Theorem 7. *The class of DTWA-recognizable languages is closed under complement.*

Proof. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, \{q_f\})$ be a DTWA that only moves to q_f at the root. The **backwards configuration graph** $\mathrm{BCG}(\mathcal{A}, t)$ is defined as a tree over $Q \times \mathrm{dom}_t$ with root (q_f, ε). For a node (q, u), the children are all (p, v) such that $(p, v) \to_{\mathcal{A}} (q, u)$. We define $\overline{\mathcal{A}}$ in a way that it performs DFS on the BCG of the input tree and accepts iff the node (q_0, ε) is found.

For that, let $\prec \subseteq (Q \times \mathrm{Dir})^2$ be an arbitrary linear order on $Q \times \mathrm{Dir}$. We set $\overline{\mathcal{A}} = (\overline{Q}, \Sigma, q_0, \overline{\delta}, q_f)$ with $\overline{Q} = \{q_0, q_f\} \cup \{\langle p, (q, d)\rangle \mid p, q \in Q, d \in \mathrm{Dir}\}$. The behavior of $\overline{\delta}$ is described below. Let $\langle q, (q', d)\rangle$ be a state.

Case 1 : In the ordering $\prec$, $(\hat{q}, \hat{d})$ is the next largest element after (q', d). (for q_0 we also consider this case with the $\prec$-minimal pair.)

Case 2 : (q', d) ☐

Theorem 8. *Let $T \subseteq T_\Sigma$. T is regular iff fcns(T) is regular.*

Proof. $\Rightarrow$ Let $\mathcal{A} = (Q, \Sigma, \Delta, F)$ be a NUTA with $T(\mathcal{A}) = T$. Wlog we assume that $\mathcal{A}$ is normalized. For every transition $\tau = (L_{a,q}, a, q) \in \Delta$, let $\mathcal{B}_{a,q} = (P_{a,q}, Q, p^0_{a,q}, \Delta_{a,q}, F_{a,q})$ be a NFA with $L(\mathcal{B}_{a,q}) = L_{a,q}$. We define $\mathcal{A}_{\mathrm{fcns}} = (Q_{\mathrm{fcns}}, \Gamma, \Delta_{\mathrm{fcns}}, F_{\mathrm{fcns}})$ so that $T(\mathcal{A}_{\mathrm{fcns}}) = \mathrm{fcns}(T)$.

- $Q_{\mathrm{fcns}} = \{q_f, q_\#\} \cup \bigcup_{a \in \Sigma, q \in Q} P_{a,q}$
- $F_{\mathrm{fcns}} = \{q_f\}$
- Δ_{fcns}:
  - $(\#, q_\#)$
  - For all $p \in \bigcup_{a \in \Sigma, q \in Q} F_{a,q}$: $(\#, p)$
  - For all $a \in \Sigma, q \in F$: $(p^0_{a,q}, q_\#, a, q_f)$
  - For all $a \in \Sigma, p \in P, p' \in P, q \in Q$ with $(p, q', p') \in \bigcup_{b \in \Sigma, q' \in Q} \Delta_{b,q'}$: $(p^0_{a,q}, p', b, p)$

Via induction on $t_1 \dots t_n$, one can show that $p \in \Delta^*_{\mathrm{fcns}}(\mathrm{fcns}(t_1 \dots t_n))$ iff there are $q_1, \dots q_n \in Q$ such that $\forall i : q_i \in \Delta^*(t_i)$ and $q_1 \dots q_n \in L(\mathcal{B}_{a,q} \text{ init } p)$.

$\Leftarrow$ Let $\mathcal{A} = (Q, \Sigma, \Delta, F)$ be an NTA with $T(\mathcal{A}) = \mathrm{fcns}(T)$. We define ☐