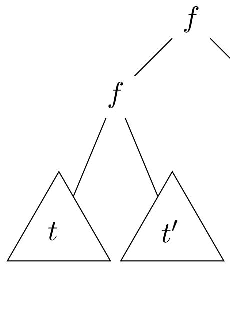
Exam

Duration: 3 hours. All paper documents permitted. The numbers $[n]$ in the margin next to questions are indications of duration and difficulty, not necessarily of the number of points you might earn from them. You must **fully justify** all your answers.

Exercise 1. Let us consider the ranked alphabet $\mathcal{F} \stackrel{\text{def}}{=} \{f^{(2)}, g^{(1)}, a^{(0)}\}$, and the family of trees $P \stackrel{\text{def}}{=} \{f(f(t,t'), g^n(a)) \mid t, t' \in T(\mathcal{F}) \wedge n > 0\}$; a tree in P is thus of the form



We define the language L of all trees that do *not* contain a tree in P as a subtree:

$$L \stackrel{\text{def}}{=} \{t \in T(\mathcal{F}) \mid \forall C \in C(\mathcal{F}).\forall t' \in T(\mathcal{F}).(t = C[t'] \Rightarrow t' \notin P)\} \ .$$

[1] 1. Give an MSO sentence ψ over $T(\mathcal{F})$ s.t. for all $t \in T(\mathcal{F})$, $t \models \psi$ if and only if $t \in L$.

We start by defining a formula $g^+(x)$ s.t. $t \models_{\nu_1 \mapsto p} g^+(x)$ if and only if the subtree $t_{|p}$ belongs to $\{g^n(a) \mid n > 0\}$ (recall that we denote implications by $\supset$):

$$g^+(x) \stackrel{\text{def}}{=} P_g(x) \wedge \forall y.x \downarrow_1^* y \supset (P_g(y) \vee P_a(y)) \ .$$

We then define a formula $P(x)$ s.t. $t \models_{\nu_1 \mapsto p} P(x)$ if and only if the subtree $t_{|p}$ belongs to P:

$$P(x) \stackrel{\text{def}}{=} P_f(x) \wedge \exists y \exists z.x \downarrow_1 y \wedge x \downarrow_2 z \wedge P_f(y) \wedge g^+(z) \ .$$

Finally, it suffices to forbid $P(x)$:

$$\psi \stackrel{\text{def}}{=} \forall x.\neg P(x) \ .$$

[1] 2. Give a PDL node formula φ s.t. for all $t \in T(\mathcal{F})$, $t, \varepsilon \models \varphi$ if and only if $t \in L$.

$$\varphi \stackrel{\text{def}}{=} \neg\langle\downarrow^*\rangle\big(f \wedge \langle\downarrow\rangle(\mathsf{first} \wedge f \wedge \langle\rightarrow; g?; (\downarrow; g?)^*; \downarrow\rangle a)\big)$$

[4] 3. Give a minimal DFTA $\mathcal{A}$ that recognises L.

It suffices to construct a minimal complete DFTA for $T(\mathcal{F}) \setminus L$ and to exchange the roles of the accepting and non-accepting states. Here is an automaton for the complement language: $\mathcal{A} \stackrel{\text{def}}{=} (Q, \mathcal{F}, \delta, \{q_P\})$ where $Q \stackrel{\text{def}}{=} \{q_a, q_+, q_f, q_g, q_P\}$ and δ is defined by

$$\begin{aligned}
a &\to q_a \\
g(q_a) &\to q_+ & g(q_+) &\to q_+ \\
g(q_f) &\to q_g & g(q_g) &\to q_g \\
g(q_P) &\to q_P \\
f(q_f, q_+) &\to q_P \\
f(q_P, q) &\to q_P & f(q, q_P) &\to q_P & &\text{(for all } q \in Q) \\
f(q, q') &\to q_f & & & &\text{(in all the other cases of } q, q')
\end{aligned}$$

The automaton $\mathcal{A}$ is deterministic and complete by definition, hence the languages of the states form a partition of $T(\mathcal{F})$.

$$\begin{aligned}
L(q_a) &= \{a\} \\
L(q_+) &= \{g^n(a) \mid n > 0\} \\
L(q_P) &= T(\mathcal{F}) \setminus L \\
L(q_g) &= \{g(t) \mid t \notin L(q_P)\} \setminus L(q_+) \\
L(q_f) &= \{f(t, t') \mid t, t' \in T(\mathcal{F})\} \cap L
\end{aligned}$$

The languages $L(q_a)$ and $L(q_+)$ are obviously correct.

A tree belongs to $L(q_P)$ iff it belongs to P and no strict subtree belongs to P (rule $f(q_f, q_+) \to q_P$) or it has a subtree that belongs to P (other rules for q_P).

A tree belongs to $L(q_g)$ iff it is of the form $g(t)$ where $t \notin L(q_a) \cup L(q_+) \cup L(q_P)$, thus since the languages form a partition of $T(\mathcal{F})$, iff $t \in L(q_g) \cup L(q_f)$, which is enforced by the rules $g(q_g) \to q_g$ and $g(q_f) \to q_g$.

Finally, a tree belongs to $L(q_f)$ iff it is rooted by f and belongs to L, and this last condition is equivalent to not belonging to $L(q_P)$, which is exactly what the rules for q_f do.

Regarding minimality, since the $(L(q))_{q \in Q}$ form a partition of $T(\mathcal{F})$, the automaton is minimal.

Alternatively, we could show that there are contexts C distinguishing some trees $t \in L(q)$ from $t' \in L(q')$, i.e. with $C[t] \in L$ and $C[t'] \in T(\mathcal{F}) \setminus L$, for all $q \neq q'$ in Q.

The empty context $\square$ distinguishes any tree $t \in L(q)$ for $q \neq q_P$ from any tree $t' \in L(q_P)$. The context $f(f(a, a), g(\square))$ distinguishes $g(f(a, a)) \in L(q_g)$ from both $a \in L(q_a)$ and $g(a) \in L(q_+)$. The context $f(f(a, a), \square)$ distinguishes $a \in L(q_a)$ from

$g(a) \in L(q_+)$. The context $f(\Box, g(a))$ distinguishes $g(f(a,a)) \in L(q_g)$, $a \in L(q_a)$, and $g(a) \in L(q_+)$ from $f(a,a) \in L(q_f)$.

Inspired by TD 6 Ex. 4

Exercise 2 (Conditional PDL)**.** We consider a fragment of PDL on an alphabet Σ called *conditional PDL*, with path formulæ restricted to the syntax

$$\alpha ::= \rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \qquad \text{(atomic paths)}$$
$$\pi ::= \alpha \mid \varphi? \mid (\alpha; \varphi?)^* \mid \pi + \pi \mid \pi; \pi \qquad \text{(conditional paths)}$$

The semantics are as in full PDL over unranked trees in $T(\Sigma)$, with the shorthands $\leftarrow \stackrel{\text{def}}{=} \rightarrow^{-1}$ and $\uparrow \stackrel{\text{def}}{=} \downarrow^{-1}$.

The logic $\mathrm{FO}(\downarrow^*, \rightarrow^*, (P_a)_{a\in\Sigma})$ is defined by the abstract syntax

$$\psi ::= P_a(x) \mid x = y \mid x \downarrow^* y \mid x \rightarrow^* y \mid \neg\psi \mid \psi \wedge \psi \mid \exists x.\psi$$

where x, y range over a countable set of first-order variables $\mathcal{X}$ and a ranges over Σ. The semantics are as usual with MSO formulæ over unranked trees in $T(\Sigma)$. To refresh your memory if you had any doubt, here are the atomic cases:

$$\begin{aligned}
&t \models_\nu P_a(x) && \text{if } t(\nu(x)) = a\ ,\\
&t \models_\nu x \downarrow^* y && \text{if } \exists p \in \mathbb{N}^*_{>0}.\nu(x) \cdot p = \nu(y)\ ,\\
&t \models_\nu x \rightarrow^* y && \text{if } \exists p \in \mathbb{N}^*_{>0}.\exists i \in \mathbb{N}_{>0}.\exists j \in \mathbb{N}.\nu(x) = p \cdot i \wedge \nu(y) = p \cdot (i+j)\ .
\end{aligned}$$

We wish to show a translation from conditional PDL into $\mathrm{FO}(\downarrow^*, \rightarrow^*, (P_a)_{a\in\Sigma})$, in the form of formulæ $\mathrm{ST}_x(\varphi)$ with a free variable x and formulæ $\mathrm{ST}_{x,y}(\pi)$ with two free variables x and y, such that, for all $t \in T(\Sigma)$ and all valuations ν,

$$t \models_\nu \mathrm{ST}_x(\varphi) \text{ iff } t, \nu(x) \models \varphi, \text{ and } t \models_\nu \mathrm{ST}_{x,y}(\pi) \text{ iff } t, \nu(x), \nu(y) \models \pi\ . \qquad (*)$$

We proceed by induction on the conditional PDL formulæ; we only consider two cases of this induction.

[1] 1. Give $\mathrm{ST}_{x,y}(\downarrow)$.

The position y is the closest strict descendent of x:

$$\mathrm{ST}_{x,y}(\downarrow) \stackrel{\text{def}}{=} x \downarrow^* y \wedge x \neq y \wedge \forall z.(x \neq z \wedge x \downarrow^* z) \supset y \downarrow^* z\ .$$

[2] 2. Give $\mathrm{ST}_{x,y}((\alpha; \varphi?)^*)$.

Let us first define $\mathrm{ST}_{x,y}(\alpha^*)$ for all atomic paths α:

$$\mathrm{ST}_{x,y}(\rightarrow^*) \stackrel{\text{def}}{=} x \rightarrow^* y\ , \qquad \mathrm{ST}_{x,y}(\leftarrow^*) \stackrel{\text{def}}{=} y \rightarrow^* x\ ,$$
$$\mathrm{ST}_{x,y}(\uparrow^*) \stackrel{\text{def}}{=} y \downarrow^* x\ , \qquad \mathrm{ST}_{x,y}(\downarrow^*) \stackrel{\text{def}}{=} x \downarrow^* y\ .$$

Then x and y must be related by $\mathrm{ST}_{x,y}(\alpha^*)$ and every position along the way except for x must satisfy φ:

$$\mathrm{ST}_{x,y}((\alpha;\varphi?)^*) \stackrel{\text{def}}{=} \mathrm{ST}_{x,y}(\alpha^*) \wedge \forall z.(x \neq z \wedge \mathrm{ST}_{x,z}(\alpha^*) \wedge \mathrm{ST}_{z,y}(\alpha^*)) \supset \mathrm{ST}_z(\varphi) \;.$$

Inspired by TD 3 Ex. 1 and TATA Thm. 1.4.4

Exercise 3 (Bottom-up Tree Transducers)**.** A *bottom-up tree transducer* (NUTT) is a tuple $\mathcal{U} = (P, \mathcal{F}, \mathcal{F}', P_f, \Delta)$ where P is a finite set of states, each state being viewed as a unary symbol, $\mathcal{F}$ and $\mathcal{F}'$ are finite ranked input and output alphabets, $P_f \subseteq P$ is a set of accepting states, and Δ is a finite set of term rewriting rules of the form $f(p_1(x_1), \ldots, p_n(x_n)) \to p(u)$ where $f \in \mathcal{F}_n$, $p, p_1, \ldots, p_n \in P$, and u is a term in $T(\mathcal{F}', \{x_1, \ldots, x_n\})$.

A NUTT $\mathcal{U}$ thus defines a rewriting system over $T(\mathcal{F} \cup \mathcal{F}' \cup P)$. The *relation* induced by $\mathcal{U}$ is then

$$R(\mathcal{U}) \stackrel{\text{def}}{=} \{(t, t') \in T(\mathcal{F}) \times T(\mathcal{F}') \mid \exists p \in P_f . t \to^*_{\mathcal{U}} p(t')\} \;.$$

[4] Show that recognisable tree languages are effectively closed under inverse NUTT transductions: given a finite tree automaton $\mathcal{A} = (Q, \mathcal{F}', \delta, Q_f)$ and a NUTT $\mathcal{U} = (P, \mathcal{F}, \mathcal{F}', P_f, \Delta)$, show how to compute an NFTA $\mathcal{A}'$ over $\mathcal{F}$ such that

$$L(\mathcal{A}') = \{t \in T(\mathcal{F}) \mid \exists t' \in L(\mathcal{A}).(t, t') \in R(\mathcal{U})\} \;.$$

You can assume $\mathcal{A}$ to be complete deterministic.

Since $\mathcal{U}$ is processing trees bottom-up, it will be convenient to also view $\mathcal{A}$ as a bottom-up DFTA, with rules of the form $f(q_1, \ldots, q_n) \to_{\mathcal{A}} q$ for $(q, f, q_1, \ldots, q_n)$ in δ.

We define $\mathcal{A}' \stackrel{\text{def}}{=} (Q \times P, \mathcal{F}, \delta', Q_f \times P_f)$ where δ' is the set of rules

$$f((q_1, p_1), \ldots, (q_n, p_n)) \to_{\mathcal{A}'} (q, p)$$

such that $n \geq 0$, $f \in \mathcal{F}_n$,

$$f(p_1, \ldots, p_n) \to_{\mathcal{U}} p(u) \in \Delta$$

for some $u \in T(\mathcal{F}', \{x_1, \ldots, x_n\})$, and

$$u\sigma \to^*_{\mathcal{A}} q \qquad (\dagger)$$

using the substitution $\sigma{:}\{x_1, \ldots, x_n\} \to Q$ defined by $\sigma(x_i) \stackrel{\text{def}}{=} q_i$ for all $1 \leq i \leq n$. Equation (†) boils down to running the DFTA $\mathcal{A}$ on u where each x_i-labelled leaf is replaced by the state $\sigma(x_i)$; the obtained state q exists and is unique since $\mathcal{A}$ is a complete DFTA.

We prove the correction of $\mathcal{A}'$ by showing by induction over $t \in T(\mathcal{F})$ that, for all $q \in Q$ and $p \in P$,

$$\exists t' \in T(\mathcal{F}') \,.\, t \to^*_{\mathcal{U}} p(t') \text{ and } t' \to^*_{\mathcal{A}} q \quad \Leftrightarrow \quad t \to^*_{\mathcal{A}'} (q, p) \;. \qquad (\ddagger)$$

base case: $t = a$ is a leaf in $\mathcal{F}_0$. By definition of δ', there is a rule $a \to_{\mathcal{U}} p(u)$ in Δ with $u \in T(\mathcal{F}')$ and $u \to^*_{\mathcal{A}} q$ if and only if $a \to_{\mathcal{A}'} (q, p)$ is a rule of δ'.

induction step: $t = f(t_1, \ldots, t_n)$ for some $n > 0$, $f \in \mathcal{F}_n$, and $t_1, \ldots, t_n \in T(\mathcal{F})$. It will be clearer to prove each implication of (‡) independently.

($\Rightarrow$) There must be a rule $f(p_1(t_1), \ldots, p_n(t_n)) \to p(t')$ with $t_i \to^*_{\mathcal{U}} p_i(t'_i)$ for all $1 \leq i \leq n$ and $t' = u[x_i \mapsto t'_i]_{1 \leq i \leq n} \to^*_{\mathcal{A}} q$. Since $\mathcal{A}$ is complete deterministic, this last rewriting can be decomposed using $t'_i \to^*_{\mathcal{A}} q_i$ for $1 \leq i \leq n$ and $u\sigma \to^*_{\mathcal{A}} q$ for the substitution $\sigma(x_i) = q_i$ for some $q_1, \ldots, q_n \in Q$.

By definition of δ, this entails that there is a rule $f((q_1, p_1), \ldots, (q_n, p_n)) \to_{\mathcal{A}'} (q, p)$. Furthermore, by induction hypothesis (‡), $t_i \to^*_{\mathcal{A}'} (q_i, p_i)$ for all $1 \leq i \leq n$. This shows $t \to^*_{\mathcal{A}'} (q, p)$.

($\Leftarrow$) There must be a rule $f((q_1, p_1), \ldots, (q_n, p_n)) \to_{\mathcal{A}'} (q, p)$ for some (q_i, p_i) such that $t_i \to^*_{\mathcal{A}'} (q_i, p_i)$ for all $1 \leq i \leq n$.

The rule implies by definition of δ that $f(p_1(x_1), \ldots, p_n(x_n)) \to_{\mathcal{U}} p(u)$ for some $u \in T(\mathcal{F}', \{x_1, \ldots, x_n\})$ such that $u[x_i \mapsto q_i]_{1 \leq i \leq n} \to^*_{\mathcal{A}} q$. Furthermore, by induction hypothesis (‡), for all $1 \leq i \leq n$ there exists t'_i such that $t_i \to^*_{\mathcal{U}} p_i(t'_i)$ and $t'_i \to^*_{\mathcal{A}} q_i$. Thus $t = f(t_1, \ldots, t_n) \to^*_{\mathcal{U}} p(u[x_i \mapsto t'_i]_{1 \leq i \leq n})$ and $u[x_i \mapsto t'_i]_{1 \leq i \leq n} \to^*_{\mathcal{A}} q$.

To conclude, note that $(t, t') \in R(\mathcal{U})$ and $t' \in L(\mathcal{A})$ if and only if $p \in P_f$ and $q \in Q_f$ in (‡), which is if and only if $t \in L(\mathcal{A}')$ as desired.

Inspired by TD 6 Ex. 1

Exercise 4 (Alternating Word Automata)**.** An *alternating word automaton* (AWA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0)$ where A is a finite set of states, Σ a finite alphabet, $\delta : Q \times \Sigma \to \mathbb{B}(Q)$ a transition function, and $q_0 \in Q$ an initial state, where $\mathbb{B}(Q)$ is the set of *positive Boolean formulæ* over Q, defined by the abstract syntax

$$\phi ::= \bot \mid \top \mid q \mid \phi \vee \phi \mid \phi \wedge \phi$$

where q ranges over Q.

An *accepting run* of $\mathcal{A}$ over a word $w = a_1 \cdots a_n$ with $n \geq 0$ and the $a_i \in \Sigma$ is an unranked tree $t \in T(Q)$ of height at most n, with root label q_0 and such that, at all its positions $p \in \operatorname{dom} t$, if $t(p) = q$ and p has m children, then $\{t(p1), \ldots, t(pm)\} \models \delta(q, a_{|p|+1})$. The language of $\mathcal{A}$ is the set of words with an accepting run.[1]

[3] 1. Given $\mathcal{A} = (Q, \Sigma, \delta, q_0)$ an AWA and $w \in \Sigma^*$, construct an NFHA $\mathcal{A}'$ that recognises exactly the set of accepting runs in $T(Q)$ of $\mathcal{A}$ on w.

[1] Note that, unlike in TD 6 Ex. 1, there is a single initial state and no accepting states; the latter are handled instead through assignments $\emptyset \models \delta(q, a)$. The definition here makes it easier to complement AWA.

Let $w = a_1 \cdots a_n$ with $a_i \in \Sigma$. We define $\mathcal{A}' \stackrel{\text{def}}{=} (Q', Q, \delta', \{(q_0, 0)\})$ where $Q' \stackrel{\text{def}}{=} Q \times \{0, \ldots, n-1\}$. The transition relation δ' has all the rules $q(R_{q,i}) \to (q, i)$ for $q \in Q$ and $0 \leq i < n$, where the horizontal language $R_{q,i}$ is defined by

$$R_{q,n-1} \stackrel{\text{def}}{=} \{\{\varepsilon\} \mid \emptyset \models \delta(q, a_n)\}$$

and for $i < n-1$

$$R_{q,i} \stackrel{\text{def}}{=} \{(q_1, i+1) \cdots (q_m, i+1) \mid \{q_1, \ldots, q_m\} \models \delta(q, a_{i+1})\}$$

Note that each $R_{q,i}$ is closed under commutation.

Each set $R_{q,i}$ is regular: one way to see this is that it is *upwards-closed* for the scattered subword ordering: if $q_1 \cdots q_m \in R_{q,i}$ and $q_1 \cdots q_m \leq q'_1 \cdots q'_{m'}$ then $\{q_1, \ldots, q_m\} \subseteq \{q'_1, \ldots, q'_{m'}\}$, and $\{q_1, \ldots, q_m\} \models \delta(q, a_{i+1})$ therefore implies $\{q'_1, \ldots, q'_{m'}\} \models \delta(q, a_{i+1})$. By Haines' Theorem, any upwards-closed set for the scattered subword ordering over a finite alphabet is regular.

If you don't know Haines' Theorem, you can nevertheless show that $R_{q,i}$ is regular by looking at the sequences $q_1 \cdots q_m$ with *no repetition* such that $\{q_1, \ldots, q_m\} \models \delta(q, a_{i+1})$. There are finitely many of these. Then a word belongs to $R_{q,i}$ if and only if it is of the form $w_0 q_1 w_1 \cdots w_m q_m w_{m+1}$ where $q_1 \cdots q_m$ is a word without repetition in $R_{q,i}$ and each of the w_i is in Q^*. This is easily described by a regular expression.

Let $t \in T(Q)$ be an accepting run of $\mathcal{A}$ on w. Let us show that $t \in L(\mathcal{A}')$ by showing that $\rho \in T(Q')$ defined by $\operatorname{dom} \rho = \operatorname{dom} t$ and $\rho(p) \stackrel{\text{def}}{=} (q, |p|)$ is an accepting run of $\mathcal{A}'$ on t. Indeed, at the root, $\rho(\varepsilon) = (q_0, 0)$ is accepting in $\mathcal{A}'$. Furthermore, at any position $p \in \operatorname{dom} \rho$, $t(p) = q$ and $\rho(p) = (q, |p|)$ are such that the sequence of children $p1, \ldots, pm$ of p form a word in $R_{q,|p|}$ since $\{t(p1), \ldots, t(pm)\} \models \delta(q, a_{|p|+1})$.

Let now $t \in T(Q)$ be a tree accepted by $\mathcal{A}'$, and $\rho \in T(Q')$ an accepting run witnessing it. Let us show that t is an accepting run of $\mathcal{A}$ on w. By definition of δ', for all positions $p \in \operatorname{dom} t$, $\rho(p) = (t(p), |p|)$. Hence at the root, $t(\varepsilon) = q_0$ since $\rho(\varepsilon) = (q_0, 0)$. Furthermore, at all positions $p \in \operatorname{dom} t$, if p has m children then $\rho(p1) \cdots \rho(pm) \in R_{q,|p|}$ implies $\{t(p1), \ldots, t(pm)\} \models \delta(q, a_{|p|+1})$. Hence t is indeed an accepting execution of $\mathcal{A}$ on w.

[4] 2. Reduce the emptiness problem in NFTA to the membership problem in AWA over the singleton alphabet $\Sigma \stackrel{\text{def}}{=} \{a\}$.

Let $\mathcal{A} = (Q, \mathcal{F}, \delta, Q_f)$ be an NFTA. By the pumping lemma (actually a pigeonhole argument suffices), if $L(\mathcal{A}) \neq \emptyset$, then there exists a tree of height at most $|Q|$ in $L(\mathcal{A})$. We build an instance $\langle \mathcal{A}', a^{|Q|} \rangle$ of the membership problem in AWA, where $\mathcal{A}' \stackrel{\text{def}}{=} (Q', \Sigma, \delta', q_0)$ where $Q' \stackrel{\text{def}}{=} Q \times \mathcal{F} \uplus \{q_0\}$. For a pair (q, f) in $Q \times \mathcal{F}_n$ with

$n \geq 0$, we define the formula

$$\delta'((q,f),a) \stackrel{\text{def}}{=} \bigvee_{(q,f,q_1,\ldots,q_n)\in\delta} \bigwedge_{i\in\{1,\ldots,n\}} \bigvee_{g\in\mathcal{F}} (q_i,g) \;.$$

Regarding the initial state q_0, we make a disjunction over all possible pairs in $Q_f \times \mathcal{F}$

$$\delta'(q_0,a) \stackrel{\text{def}}{=} \bigvee_{q\in Q_f} \bigvee_{f\in\mathcal{F}} \delta'((q,f),a) \;.$$

Let us assume that $L(\mathcal{A})$ is not empty. Then there exists a tree $t \in L(\mathcal{A})$ with height at most $|Q|$, i.e. $|p| < |Q|$ for all $p \in \operatorname{dom} t$. Thus there exists an accepting run $\rho \in T(Q)$ with $\operatorname{dom}\rho = \operatorname{dom} t$, $\rho(\varepsilon) \in Q_f$, and every elementary tree in ρ is consistent with δ.

We map the pair t,ρ to a tree t' in $T(Q \times \{0,\ldots,|Q|-1\})$ with domain $\operatorname{dom} t' \stackrel{\text{def}}{=} \operatorname{dom} t$,

$$t'(\varepsilon) \stackrel{\text{def}}{=} (q_0,0)\;, \qquad\qquad t'(p) \stackrel{\text{def}}{=} (q(p),t(p))$$

for all $p \in \operatorname{dom} t \setminus \{\varepsilon\}$. It remains to show that t' is an accepting run of $\mathcal{A}'$ on $a^{|Q|}$. Let p be a position in $\operatorname{dom} t'$ other than ε. Then $t(p) = f$ and $\rho(p) = q$ for some $f \in \mathcal{F}_n$ for some n and $q \in Q$. Since ρ is a run of $\mathcal{A}$, p has n children $p1,\ldots,pn$ and there exists a transition $(q,f,q_1,\ldots,q_n) \in \delta$ with $\rho(pi) = q_i$ for all $1 \leq i \leq n$; let also $t(pi) = g_i \in \mathcal{F}$ for all $1 \leq i \leq n$. Thus $\{(q_1,g_1),\ldots,(q_n,g_n)\} \models \delta'((q,f),a)$. Finally, for $p = \varepsilon$, there exists $\rho(\varepsilon) \in Q_f$ and $t(\varepsilon) \in \mathcal{F}_n$ for some n such that, in a similar way, the children satisfy $\{(\rho(1),t(1)),\ldots,(\rho(n),t(n))\} \models \delta'((\rho(\varepsilon),t(\varepsilon)),a)$, thus $\{(\rho(1),t(1)),\ldots,(\rho(n),t(n))\} \models \delta'(q_0,a)$ as desired.

Conversely, assume there is an accepting run $t' \in T(Q)$ of $\mathcal{A}'$. Let us argue that every position $p \neq \varepsilon$ of t' with $t'(p) = q$ can be assumed to have exactly n children for some $(q,f,q_1,\ldots,q_n) \in \delta$ with $\forall 1 \leq i \leq n$, $t'(pi) = (q_i,g_i)$ and $g_i \in \mathcal{F}$. Indeed, its list of children must satisfy $q_1 \wedge \cdots \wedge q_n$ for some $(q,f,q_1,\ldots,q_n) \in \delta$, hence this list must contain at least one occurrence of every state in $\{q_1,\ldots,q_n\}$. If pi is a child position with $t(pi) \notin \{q_1,\ldots,q_n\}$, then that entire subtree can be removed and we still have an accepting tree. For a state $q \in \{q_1,\ldots,q_n\}$, if its number of occurrences among the children of p is larger than its number of occurrences in $q_1,\ldots,q_n$, then we can similarly remove some subtrees rooted by q and still obtain an accepting run. Conversely, if it is smaller, then there is a child position pi with $t'(pi) = q$, and that subtree can be duplicated as many times as needed to yield another accepting tree where the numbers of occurrences coincide. At this point, we know that the list of child labels of p can be assumed to be a permutation of $q_1,\ldots,q_n$. Finally, the children can be reordered such that $t'(pi) = q_i$ since the ordering does not matter for acceptance. We can assume in the same way that the list of children of the root is such that $t(i) = (q_i, g_i)$ for some $(q, f, q_1, \ldots, q_n) \in \delta$, $f \in \mathcal{F}$, and $q \in Q_f$. This is all standard reasoning on alternating automata; you could make these assumptions without justification.

Under these assumptions on t', we exhibit a tree $t \in T(\mathcal{F})$ and an accepting run $\rho \in T(Q)$ of $\mathcal{A}$ on t, thereby showing that $L(\mathcal{A})$ is not empty. We define for this $\operatorname{dom} t \stackrel{\text{def}}{=} \operatorname{dom} t'$ and for all $p \neq \varepsilon$, $t(p) \stackrel{\text{def}}{=} f \in \mathcal{F}_n$ for some n and $\rho(p) \stackrel{\text{def}}{=} q$ if $t'(p) = (q, f)$. By definition of δ', for any such position p, there exists a transition $(q, f, q_1, \ldots, q_n) \in \delta$ such that $\rho(pi) = q_i$ for all $1 \leq i \leq n$. Regarding the root ε, we also know that there exists $q \in Q_f$ and $f \in \mathcal{F}_n$ for some n such that there is a transition $(q, f, q_1, \ldots, q_n) \in \delta$ with $\rho(i) = q_i$ for all $1 \leq i \leq n$, thus defining $t(\varepsilon) \stackrel{\text{def}}{=} f$ and $\rho(\varepsilon) \stackrel{\text{def}}{=} q$ ensures that t is a tree in $L(\mathcal{A})$ with an accepting run ρ.