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 justify all your answers.

Exercise 1 (First-Order Logic with Transitive Reflexive Relations). We consider the first-order logic \(\mathcal{FO}(\downarrow^*, (P_a)_{a\in\Sigma})\) over finite unranked trees labelled by some finite alphabet \(\Sigma\) along with the descendant relation \(\downarrow^*\).

1. Give a closed first-order formula \(\psi_1\) enforcing that the sequence of labels along any branch is in \((ab)^+\). \(\text{Hint: You can use the following first-order formulae:}\)

\[
\begin{align*}
x \downarrow^+ y &\overset{\text{def}}{=} x \downarrow^* y \land x \neq y, &\quad x \downarrow y &\overset{\text{def}}{=} x \downarrow^+ y \land \exists z(x \downarrow^+ z \land z \downarrow^+ y), \\
\text{root}(x) &\overset{\text{def}}{=} \neg \exists y(y \downarrow^+ x), &\quad \text{leaf}(x) &\overset{\text{def}}{=} \neg \exists y(x \downarrow^+ y).
\end{align*}
\]

\[
\psi_1 \overset{\text{def}}{=} \exists x (P_a(x) \land \text{root}(x)) \\
\quad \land \forall x (P_a(x) \Rightarrow \exists y (P_b(y) \land x \downarrow y)) \\
\quad \land \forall y (P_b(y) \Rightarrow \text{leaf}(y) \lor \exists x (P_a(x) \land y \downarrow x)).
\]

2. Give a closed first-order formula \(\psi_2\) enforcing that every branch starting from an \(a\)-labelled position contains a \(b\)-labelled position. \(\text{Hint: You can use the following first-order formula:}\)

\[
\text{branch}(x, y) \overset{\text{def}}{=} x \downarrow^* y \land \text{leaf}(y).
\]

\[
\psi_2 \overset{\text{def}}{=} \forall x \forall y ((\text{branch}(x, y) \land P_a(x)) \Rightarrow \exists z (x \downarrow^+ z \land z \downarrow^* y \land P_b(z))).
\]

3. Let \(\Sigma = \{a, b, c\}\) and consider the formula

\[
\psi \overset{\text{def}}{=} \forall x \forall z ((P_a(x) \land x \neq z \land \text{branch}(x, z)) \\
\Rightarrow \exists y (x \downarrow^+ y \land y \downarrow^* z \land P_c(y) \land \forall z (x \downarrow^+ z \land z \downarrow^+ y \Rightarrow P_b(z)))).
\]

(a) Give an equivalent PDL node formula.

\[
[\downarrow^*] (a \Rightarrow [(\downarrow^* b?^*) \lor (a \lor (b \land \text{leaf})?^*))] \perp
\]

1
(b) Give a complete deterministic (bottom-up) finite hedge automaton for the set of models of $\psi$.

Let $Q \overset{\text{def}}{=} \{ q_\perp, q_a, q_c \}$ and $Q_f \overset{\text{def}}{=} \{ q_a, q_c \}$. The intuition is for

- $t \rightarrow^* q_\perp$ iff $t \not\models \psi$,
- $t \rightarrow^* q_a$ iff $t \models \psi$ and there is a branch with label in $b^* a \Sigma^* + b^+$, and
- $t \rightarrow^* q_c$ if $t \models \psi$ and every branch has a prefix in $b^* c$.

We use regular expressions over $Q$ to describe the horizontal languages in the rules of $\Delta$:

$$a(q_c^*) \rightarrow q_a \quad a(Q^* \cdot (q_a + q_\perp) \cdot Q^*) \rightarrow q_\perp$$

$$b(q_c^*) \rightarrow q_c \quad b(\varepsilon + (q_a + q_c)^* \cdot q_a \cdot (q_a + q_c)^*) \rightarrow q_a \quad b(Q^* \cdot q_\perp \cdot Q^*) \rightarrow q_\perp$$

$$c((q_a + q_c)^*) \rightarrow q_c \quad c(Q^* \cdot q_\perp \cdot Q^*) \rightarrow q_\perp$$

**Exercise 2** (Propositional Dynamic Logic). We work with unranked trees over a finite alphabet $\Sigma$.

1. We write $p \prec p'$ for two positions $p$ and $p'$ of a tree $t \in T(\Sigma)$ if $p$ is visited before $p'$ in a pre-order traversal of $t$. (Hence $\prec$ is a total order on $\text{Pos}(t)$).

   Define a PDL path formula $\pi$ such that $\sem{\pi}_t = \{ (p, p') \in \text{Pos}(t) \times \text{Pos}(t) \mid p \prec p' \}$ for all $t \in T(\Sigma)$.

   We start by defining a path formula for successors in a pre-order traversal, and then take its transitive closure:

   $$\text{succ} \overset{\text{def}}{=} (\downarrow; \text{first}?) + (\text{leaf}?; (\text{last}?; \uparrow)^*; \rightarrow)$$

   $$\pi \overset{\text{def}}{=} \text{succ}^+$$

2. Define a PDL path formula $\pi'$ such that $\sem{(\pi')^*}_t = \{ (p, p') \in \text{Pos}(t) \times \text{Pos}(t) \mid t(p) = t(p') \}$ and $\sem{\pi'}_t$ is a function for all $t \in T(\Sigma)$.

   We build a new path formula on top of $\text{succ}$, which wraps around the root when we reach the rightmost leaf of $t$:

   $$\text{lastleaf} \overset{\text{def}}{=} [\text{succ}]_\perp$$

   $$\text{wrap} \overset{\text{def}}{=} \text{succ} + (\text{lastleaf}?; \uparrow^*; \text{root}?)$$

   $$\pi' \overset{\text{def}}{=} \sum_{a \in \Sigma} a?; \text{wrap}; (\neg a?; \text{wrap})^*; a?$$
**Exercise 3** (Deterministic Top-Down Tree Automata). Let $t$ be a tree in $T(\mathcal{F})$ for some finite ranked alphabet $\mathcal{F}$ with maximal arity $k$, and let $\Pi \overset{\text{def}}{=} (\bigcup_{1 \leq n \leq k} \mathcal{F}_n \times \{1, \ldots, n\})^* \cdot \mathcal{F}_0$. The path language $\text{Paths}(t) \subseteq \Pi$ is defined by

$$
\text{Paths}(a) \overset{\text{def}}{=} \{a\} \quad \text{if } a \in \mathcal{F}_0 \text{ is a constant,}
$$

$$
\text{Paths}(f(t_1, \ldots, t_n)) \overset{\text{def}}{=} \bigcup_{1 \leq i \leq n} \{(f, i)\} \cdot \text{Paths}(t_i) \quad \text{if } f \in \mathcal{F}_n \text{ for some } 1 \leq n \leq k.
$$

We lift this to $\text{Paths}(L) \overset{\text{def}}{=} \bigcup_{t \in L} \text{Paths}(t)$ for any $L \subseteq T(\mathcal{F})$.

1. Show that if $L \subseteq T(\mathcal{F})$ is recognisable, then $\text{Paths}(L)$ is recognisable over the alphabet $\Sigma \overset{\text{def}}{=} \mathcal{F}_0 \cup \bigcup_{1 \leq n \leq k} \mathcal{F}_n \times \{1, \ldots, n\}$.

**Hint:** Start with a co-accessible top-down NFTA for $L$.

Let $\mathcal{A} = (Q, \mathcal{F}, Q_f, \Delta)$ be a top-down NFTA with $L(\mathcal{A}) = L$. Without loss of generality, $\mathcal{A}$ is co-accessible: $\forall q \in Q$, $\exists t \in T(\mathcal{F})$, $q \rightarrow^*_\mathcal{A} t$.

We construct $\mathcal{A}' = (Q', \Sigma, \delta, I, F)$ a NFA with state set $Q' \overset{\text{def}}{=} Q \cup \{q_\ell\}$, initial state set $I \overset{\text{def}}{=} Q_f$, accepting state set $F \overset{\text{def}}{=} \{q_\ell\}$, and transition set

$$
\delta \overset{\text{def}}{=} \{(q, (f, i), q_i) \mid 0 < i \leq n \leq k, f \in \mathcal{F}_n, (q \rightarrow f(q_1, \ldots, q_n) \in \Delta) \}
$$

$$
\cup \{(q, a, q_\ell) \mid a \in \mathcal{F}_0, (q \rightarrow a) \in \Delta\}.
$$

We denote by $L_q(\mathcal{A}') \overset{\text{def}}{=} \{w \in \Sigma^* \mid q \rightarrow^*_\mathcal{A}' q_\ell\}$ the word language recognised by $q \in Q$ in $\mathcal{A}'$; then $L(\mathcal{A}') = \bigcup_{q \in I} L_q(\mathcal{A}')$.

**Paths($L$) $\subseteq$ Paths($\mathcal{A}'$):** We prove by induction on $t \in T(\mathcal{F})$ that, if $q \rightarrow^*_\mathcal{A} t$, then $\text{Paths}(t) \subseteq L_q(\mathcal{A}')$. Thus, if $t \in L$, then $q \in Q_f = I$, and $\text{Paths}(t) \subseteq L(\mathcal{A}')$.

- For the base case where $t = a \in \mathcal{F}_0$, $a \rightarrow q$ implies $(q, a, q_\ell) \in \delta$ and thus $a \in L_q(\mathcal{A}')$.

- For the induction step where $t = f(t_1, \ldots, t_n)$, $f \in \mathcal{F}_n$ for some $1 \leq n \leq k$, we have the reduction $q \rightarrow \mathcal{A} f(q_1, \ldots, q_n) \rightarrow^*_\mathcal{A} t$ for some $f(q_1, \ldots, q_n) \rightarrow q$ in $\Delta$. We apply the induction hypothesis on each $q_i \rightarrow^*_\mathcal{A} t_i$ for $1 \leq i \leq n$:

$$
\text{Paths}(f(t_1, \ldots, t_n)) = \bigcup_{1 \leq i \leq n} \{(f, i)\} \cdot \text{Paths}(t_i) \quad \text{by def.}
$$

$$
\subseteq \bigcup_{1 \leq i \leq n} \{(f, i)\} \cdot L_q(\mathcal{A}') \quad \text{by ind. hyp.}
$$

$$
\subseteq L_q(\mathcal{A}') \quad \forall 1 \leq i \leq n, (q, (f, i), q_i) \in \delta.
$$

**L($\mathcal{A}'$) $\subseteq$ Paths($L$):** First note that $L(\mathcal{A}') \subseteq \Pi$. We show by induction on $w \in \Pi$ that, for all $q \in Q$, if $w \in L_q(\mathcal{A}')$, then there exists $t \in T(\mathcal{F})$ such that $w \in \text{Paths}(t)$ and $q \rightarrow^*_\mathcal{A} t$. Then, $w \in L(\mathcal{A}')$ occurs when $q \in I = Q_f$ and thus there exists $t \in L$ with $w \in \text{Paths}(t)$.
2. The path closure of a word language \( L \subseteq \Pi \) is

\[
\overline{L} \overset{\text{def}}{=} \{ t \in T(\mathcal{F}) \mid \text{Paths}(t) \subseteq L \}.
\]

Show that if \( L \subseteq \Pi \) is recognisable, then \( \overline{L} \subseteq T(\mathcal{F}) \) is recognisable by a deterministic top-down tree automaton.

Let \( \mathcal{A}' = \langle Q', \Sigma, \delta, I, F \rangle \) be a DFA recognising \( L \subseteq \Pi \). Observe that \( L \) is prefix: if \( ww' \in L \) and \( w, w' \in \Sigma^* \), then \( w' = \varepsilon \); this is because the symbols of \( F_0 \) act as end-of-word markers. Thus without loss of generality, \( F \) is a singleton \( \{q_0\}\).

We construct a deterministic top-down tree automaton \( \mathcal{A} = \langle Q, \mathcal{F}, Q_f, \Delta \rangle \) with

\[
Q = Q' \setminus \{q_0\}, \quad Q_f = I, \quad \text{and}
\]

\[
\Delta = \{ q \to a \mid a \in F_0, \delta(q, a) = q_0 \}
\]

\[
\cup \{ q \to f(\delta(q, (f, 1)), \ldots, \delta(q, (f, n))) \mid f \in F_n \text{ and } \forall 1 \leq i \leq n, \delta(f, i) \text{ is defined} \}.
\]

We show by induction on \( t \in T(\mathcal{F}) \) that, for all \( q \in Q \), \( \text{Paths}(t) \subseteq L_q(\mathcal{A}') \), if and only if \( q \to^*_\mathcal{A} t \). Then, \( t \in \overline{L} \), if and only if \( \text{Paths}(t) \subseteq L \), if and only if \( \text{Paths}(t) \subseteq L_q(\mathcal{A}') \) for some \( q \in I \), if and only if \( q \to^*_\mathcal{A} t \) for some \( q \in Q_f \), if and only if \( t \in L(\mathcal{A}) \).

- For the base case \( t = a \in F_0 \), \( \text{Paths}(a) = \{a\} \subseteq L_q(\mathcal{A}') \) if and only if \( \delta(q, a) = q_0 \), if and only if \( (q \to a) \in \delta \) as desired.

- For the induction step, let \( t = f(t_1, \ldots, t_n) \) for some \( 1 \leq n \leq k, f \in F_n \), and each \( t_i \in T(\mathcal{F}) \). Then \( \text{Paths}(t) = \bigcup_{1 \leq i \leq n} \{(f, i)\} \cdot \text{Paths}(t_i) \subseteq L_q(\mathcal{A}') \) if and only if \( \text{Paths}(t_i) \subseteq L_{\delta(q, (f, i))}(\mathcal{A}') \) for all \( 1 \leq i \leq n \). By induction hypothesis, this is if and only if \( \delta(q, (f, i)) \to^*_\mathcal{A} t_i \) for each \( i \), which is if and only if \( q \to^*_\mathcal{A} f(\delta(q, (f, 1)), \ldots, \delta(q, (f, n))) = t \) as desired.
3. Deduce that $L \subseteq T(F)$ is recognisable by a deterministic top-down tree automaton if and only if $L$ is recognisable and path closed, i.e. $L = \overline{\text{Paths}(L)}$.

If $L$ is recognisable by a deterministic top-down tree automaton $A$, then $L$ is recognisable and the automaton $A'$ constructed in Question 1 for $L' = \overline{\text{Paths}(L)}$ is a DFA. If we apply the construction of Question 2 to $A'$ we obtain $A$ back! Hence $L = L(A) = \overline{\text{Paths}(L)}$.

Conversely, if $L$ is recognisable and path closed, then by Question 1 $\text{Paths}(L)$ is recognised by a word automaton $A'$, which we can determinise to obtain by Question 2 a deterministic top-down tree automaton for $\overline{\text{Paths}(L)} = L$.

4. Show that it is decidable whether a recognisable tree language is path closed.

This is clearly decidable since by questions 1 and 2 we can build a deterministic top-down tree automaton of exponential size with $L(A_d) = \text{Paths}(L)$. As $L \subseteq \text{Paths}(L)$ always holds, it suffices to check whether $L(A_d) \subseteq L(A)$, i.e. whether $L(A) \cap (T(F) \setminus L(A_d)) = \emptyset$. Observe that complementing $A_d$ is trivial, hence this last inclusion test is in polynomial time in the size of $A$ and $A_d$, hence in EXP overall.

5. Let $F = \{\land(2), \lor(2), \bot(0), \top(0)\}$ and $L = \{t \in T(F) \mid e(t) = \top\}$ be the set of trees that evaluate to $\top$ according to:

$$e(\land(t_1, t_2)) = e(t_1) \land e(t_2), \quad e(\lor(t_1, t_2)) = e(t_1) \lor e(t_2), \quad e(\bot) = \bot, \quad e(\top) = \top.$$ 

Show that $L$ is not recognised by any deterministic top-down tree automaton.

Indeed, $t_1 \equiv \lor(\bot, \top)$ and $t_2 \equiv \lor(\top, \bot)$ are in $L$. Thus $(\lor, 1)\bot \in \text{Paths}(t_1)$ and $(\lor, 2)\bot \in \text{Paths}(t_2)$ show that $t_3 = \lor(\bot, \bot) \in \overline{\text{Paths}(L)}$ although it does not belong to $L$.

6. Show that $L$ is not recognised by any finite union of deterministic top-down tree automata.

Assume $L = \bigcup_{1 \leq i \leq n} L_i$ where each $L_i$ is recognised by a deterministic top-down tree automaton. Consider the trees

$$t_0 = \lor(\top, \bot), \quad t_{m+1} = \lor(\bot, t_m).$$

As all of these infinitely many trees belong to $L$, there must be $1 \leq i \leq n$ such that $t_j \in L_i$ and $t_k \in L_i$ for $j < k$. Hence $(\lor, 2)^{j-1}(\lor, 2)\bot \in \text{Paths}(t_j) \subseteq \text{Paths}(L_i)$ and $(\lor, 2)^{j}((\lor, 1)\bot \in \text{Paths}(t_k) \subseteq \text{Paths}(L_i)$ for all $\ell < j$ imply that $t'_j \in L_i = \text{Paths}(L_i)$

where

$$t'_0 = \lor(\bot, \bot), \quad t'_{m+1} = \lor(\bot, t'_m).$$

This contradicts $L_i \subseteq L$. 

5