MPRI 2-27-1 Exam

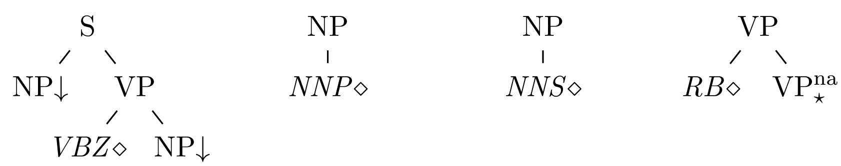
Duration: 3 hours
Paper documents are allowed. The numbers in front of questions are indicative of hardness or duration.

1 Two-level Syntax

Exercise 1 (Derivation trees)**.** In a tree adjoining grammar $\mathcal{G} = \langle N, \Sigma, T_\alpha, T_\beta, S\rangle$, the trees in $L_T(\mathcal{G})$ are called *derived* trees. We are interested here in another tree structure, called a *derivation* tree, for which we propose a formalisation here. Let us assume for simplicity that all the foot nodes of auxiliary trees have the '$^{\mathrm{na}}$' null adjunction annotation.

For an elementary tree $\gamma \in T_\alpha \uplus T_\beta$, we define its *contents* $c(\gamma)$ to be a finite sequence over the alphabet $Q \stackrel{\text{def}}{=} \{q_A \mid A \in N \uplus N{\downarrow}\}$. Formally, we enumerate for this the labels in Q of its nodes in position order; the nodes labelled by $\Sigma \cup N^{\mathrm{na}}$ are ignored.

Consider for instance the TAG $\mathcal{G}_1$ with $N \stackrel{\text{def}}{=} \{\mathrm{S}, \mathrm{NP}, \mathrm{VP}\}$, $\Sigma \stackrel{\text{def}}{=} \{\mathit{VBZ}\diamond, \mathit{NNP}\diamond, \mathit{NNS}\diamond, \mathit{RB}\diamond\}$, $T_\alpha \stackrel{\text{def}}{=} \{\mathit{likes}, \mathit{Bill}, \mathit{mushrooms}\}$, $T_\beta \stackrel{\text{def}}{=} \{\mathit{possibly}\}$, and $S \stackrel{\text{def}}{=} \mathrm{S}$, where the elementary trees are shown below:



(*likes*) (*Bill*) (*mushrooms*) (*possibly*)

Then *likes* has contents $c(\mathit{likes}) = q_{\mathrm{S}}, q_{\mathrm{NP}\downarrow}, q_{\mathrm{VP}}, q_{\mathrm{NP}\downarrow}$, $c(\mathit{Bill}) = q_{\mathrm{NP}}$, $c(\mathit{mushrooms}) = q_{\mathrm{NP}}$, and $c(\mathit{possibly}) = q_{\mathrm{VP}}$.

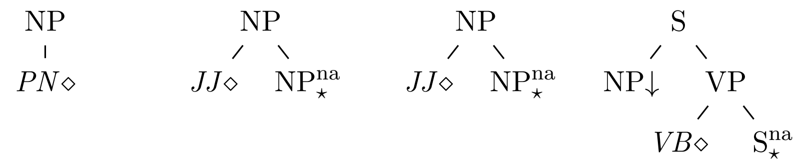
We now define a finite ranked alphabet $\mathcal{F} \stackrel{\text{def}}{=} T_\alpha \uplus T_\beta \uplus \{\varepsilon^{(0)}\}$. For an elementary tree $\gamma \in T_\alpha \uplus T_\beta$, its *rank* is $r(\gamma) \stackrel{\text{def}}{=} |c(\gamma)|$ the length of its contents. For the symbol ε, its rank is $r(\varepsilon) \stackrel{\text{def}}{=} 0$. For a TAG $\mathcal{G} = \langle N, \Sigma, T_\alpha, T_\beta, S\rangle$, we construct a finite tree automaton $\mathcal{A}_{\mathcal{G}} \stackrel{\text{def}}{=} \langle Q, \mathcal{F}, \delta, q_{S\downarrow}\rangle$ where Q and $\mathcal{F}$ are defined as above and

$$\begin{aligned}\delta \stackrel{\text{def}}{=} &\{(q_{A\downarrow}, \alpha^{(r(\alpha))}, c(\alpha)) \mid A{\downarrow} \in N{\downarrow}, \alpha \in T_\alpha, \mathrm{rl}(\alpha) = A\}\\ &\cup \{(q_A, \beta^{(r(\beta))}, c(\beta)) \mid A \in N, \beta \in T_\beta, \mathrm{rl}(\beta) = A\}\\ &\cup \{(q_A, \varepsilon^{(0)}) \mid A \in N\}\end{aligned}$$

where 'rl' returns the root label of the tree.

[1] 1. Give the finite automaton $\mathcal{A}_{\mathcal{G}_1}$ associated with the example TAG $\mathcal{G}_1$.

[1] 2. Modify your automaton in order to also handle the trees *someone* $\in T_\alpha$ and *real*, *fake*, *thinks* $\in T_\beta$ shown below, where $PN\diamond$, $JJ\diamond$, $VB\diamond \in \Sigma$:



(*someone*) (*real*) (*fake*) (*thinks*)

[1] 3. The intention that our finite automaton generates the *derivation* language $L_D(\mathcal{G}) \stackrel{\text{def}}{=} L(\mathcal{A}_\mathcal{G})$ of $\mathcal{G}$. Can you figure out what should be the derivation tree of '*Someone possibly thinks Bill likes mushrooms*'?

[2] 4. Give a PDL node formula φ_1 such that $L(\mathcal{A}_{\mathcal{G}_1}) = \{t \in T(\mathcal{F}) \mid t, \text{root} \models \varphi_1\}$.

1.1 Macro Tree Transducers

Let $\mathcal{X}$ be a countable set of variables and $\mathcal{Y}$ a countable set of parameters; we assume $\mathcal{X}$ and $\mathcal{Y}$ to be disjoint. For Q a ranked alphabet with arities greater than zero, we abuse notations and write $Q(\mathcal{X})$ for the alphabet of pairs $(q, x) \in Q \times \mathcal{X}$ with $\mathit{arity}(q, x) \stackrel{\text{def}}{=} \mathit{arity}(q) - 1$. This is just for convenience, and $(q, x)(t_1, \ldots, t_n)$ is really the term $q(x, t_1, \ldots, t_n)$.

Syntax. A *macro tree transducer* (NMTT) is a tuple $\mathcal{M} = (Q, \mathcal{F}, \mathcal{F}', \Delta, I)$ where Q is a finite set of states, all of arity ≥ 1, $\mathcal{F}$ and $\mathcal{F}'$ are finite ranked alphabets, $I \subseteq Q_1$ is a set of root states of arity one, and Δ is a finite set of term rewriting rules of the form $q(f(x_1, \ldots, x_n), y_1, \ldots, y_p) \to e$ where $q \in Q_{1+p}$ for some $p \geq 0$, $f \in \mathcal{F}_n$ for some $n \in \mathbb{N}$, and $e \in T(\mathcal{F}' \cup Q(\mathcal{X}_n), \mathcal{Y}_p)$. Note that this imposes that any occurrence in e of a variable $x \in \mathcal{X}$ must be as the first argument of a state $q \in Q$.

Inside-Out Semantics. Given a NMTT, the *inside-out* rewriting relation over trees in $T(\mathcal{F} \cup \mathcal{F}' \cup Q)$ is defined by: $t \xrightarrow{\text{IO}} t'$ if there exist a rule $q(f(x_1, \ldots, x_n), y_1, \ldots, y_p) \to e$ in Δ, a context $C \in C(\mathcal{F} \cup \mathcal{F}' \cup Q)$, and two substitutions $\sigma\colon \mathcal{X} \to T(\mathcal{F})$ and $\rho\colon \mathcal{Y} \to T(\mathcal{F}')$ such that $t = C[q(f(x_1, \ldots, x_n), y_1, \ldots, y_p)\sigma\rho]$ and $t' = C[e\sigma\rho]$. In other words, in inside-out rewriting, when applying a rewriting rule $q(f(x_1, \ldots, x_n), y_1, \ldots, y_p) \to e$, the parameters $y_1, \ldots, y_p$ must be mapped to trees in $T(\mathcal{F}')$, with no remaining states from Q.

Similarily to context-free tree grammars, the *inside-out* transduction $[\![\mathcal{M}]\!]_{\text{IO}}$ realised by $\mathcal{M}$ is defined through inside-out rewriting semantics:

$$[\![\mathcal{M}]\!]_{\text{IO}} \stackrel{\text{def}}{=} \{(t, t') \in T(\mathcal{F}) \times T(\mathcal{F}') \mid \exists q \in I \,.\, q(t) \xrightarrow{\text{IO}}{}^{*} t'\} \,.$$

Example 1. Let $\mathcal{F} \stackrel{\text{def}}{=} \{a^{(1)}, \$^{(0)}\}$ and $\mathcal{F}' \stackrel{\text{def}}{=} \{f^{(3)}, a^{(1)}, b^{(1)}, \$^{(0)}\}$. Consider the NMTT $\mathcal{M} = (\{q^{(1)}, q'^{(3)}\}, \mathcal{F}, \mathcal{F}', \Delta, \{q\})$ with Δ the set of rules

$$q(a(x_1)) \to q'(x_1, \$, \$) \qquad q'(\$, y_1, y_2) \to f(y_1, y_1, y_2)$$
$$q'(a(x_1), y_1, y_2) \to q'(x_1, a(y_1), a(y_2)) \qquad q'(a(x_1), y_1, y_2) \to q'(x_1, a(y_1), b(y_2))$$
$$q'(a(x_1), y_1, y_2) \to q'(x_1, b(y_1), a(y_2)) \qquad q'(a(x_1), y_1, y_2) \to q'(x_1, b(y_1), b(y_2))$$

Then we have for instance the following derivation:

$$\begin{aligned} q(a(a(a(\$)))) &\xrightarrow{\text{IO}} q'(a(a(\$)), \$, \$) \\ &\xrightarrow{\text{IO}} q'(a(\$), b(\$), b(\$)) \\ &\xrightarrow{\text{IO}} q'(\$, a(b(\$)), b(b(\$))) \\ &\xrightarrow{\text{IO}} f(a(b(\$)), a(b(\$)), b(b(\$))) \end{aligned}$$

showing that $(\ a(a(a(\$))),\ f(a(b(\$)), a(b(\$)), b(b(\$)))\ ) \in [\![\mathcal{M}]\!]$.

Exercise 2 (Monadic trees)**.** An NMTT $\mathcal{M}$ is called *linear* and *non-deleting* if, in every rule $q(f(x_1, \ldots, x_n), y_1, \ldots, y_p) \to e$ in Δ, the term e is linear in $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_p\}$, i.e. each variable and each parameter occurs exactly once in the term e.

Let $\mathcal{F}' \stackrel{\text{def}}{=} \{a^{(1)}, b^{(1)}, \$^{(0)}\}$. Observe that trees in $T(\mathcal{F}')$ are in bijection with contexts in $C(\mathcal{F}')$ and words over $\{a, b\}^*$. For a context C from $C(\mathcal{F}')$, we write C^R for its *mirror context*, read from the leaf to the root. For instance, if $C = a(b(a(a(\Box))))$, then $C^R = a(a(b(a(\Box))))$. Formally, let $n \in \mathbb{N}$ be such that $\operatorname{dom} C = \{0^m \mid m \leq n\}$; then $C(0^n) = \Box$ and $C(0^m) \in \{a, b\}$ for $m < n$. Then C^R is defined by $\operatorname{dom} C^R \stackrel{\text{def}}{=} \operatorname{dom} C$, $C^R(0^n) \stackrel{\text{def}}{=} \Box$, and $C^R(0^m) \stackrel{\text{def}}{=} C^R(0^{n-m})$ for all $m < n$.

[2] 1. Give a linear and non-deleting NMTT $\mathcal{M}$ from $\mathcal{F}'$ to $\mathcal{F}'$ such that $[\![\mathcal{M}]\!]_{\text{IO}} = \{(C[\$], C[C^R[\$]]) \mid C \in C(\mathcal{F}')\}$. In terms of words over $\{a, b\}^*$, this transducer maps w to the palindrome ww^R. Is $[\![\mathcal{M}]\!]_{\text{IO}}(T(\mathcal{F}))$ a recognisable tree language?

Exercise 3 (From derivation to derived trees)**.** Consider again the tree adjoining grammar $\mathcal{G}_1$ from Exercise 1.

[3] 1. Give a linear non-deleting NMTT $\mathcal{M}_1$ that maps the derivation trees of $\mathcal{G}_1$ to its derived trees. Formally, we want $\operatorname{dom}([\![\mathcal{M}_1]\!]_{\text{IO}}) = L_D(\mathcal{G}_1)$ and $[\![\mathcal{M}_1]\!]_{\text{IO}}(T(\mathcal{F})) = L_T(\mathcal{G}_1)$.

Exercise 4 (Context-free tree grammar)**.** Let $\mathcal{M} = (Q, \mathcal{F}, \mathcal{F}', \Delta, I)$ be an NMTT and $\mathcal{A} = (Q', \mathcal{F}, \delta, I')$ be an NFTA.

[5] 1. Show that $L \stackrel{\text{def}}{=} [\![\mathcal{M}]\!]_{\text{IO}}(L(\mathcal{A})) = \{t' \in T(\mathcal{F}') \mid \exists t \in L(\mathcal{A})\,.\,(t, t') \in [\![\mathcal{M}]\!]_{\text{IO}}\}$ is an inside-out context-free tree language, i.e., show how to construct a CFTG $\mathcal{G} = (N, \mathcal{F}', S, R)$ such that $L_{\text{IO}}(\mathcal{G}) = L$.

2 Scope ambiguities and propositional attitudes

Exercise 5. One considers the two following signatures:

(Σ_{ABS})

$$\begin{aligned}
\textsc{suzy} &: NP \\
\textsc{bill} &: NP \\
\textsc{mushroom} &: N \\
\textsc{a} &: N \to (NP \to S) \to S \\
\textsc{a}_{\mathit{inf}} &: N \to (NP \to S_{\mathit{inf}}) \to S_{\mathit{inf}} \\
\textsc{eat} &: NP \to NP \to S_{\mathit{inf}} \\
\textsc{to} &: (NP \to S_{\mathit{inf}}) \to VP \\
\textsc{want} &: VP \to NP \to S
\end{aligned}$$

$(\Sigma_{\text{S-FORM}})$

$$\begin{aligned}
\boldsymbol{Suzy} &: \mathit{string} \\
\boldsymbol{Bill} &: \mathit{string} \\
\boldsymbol{mushroom} &: \mathit{string} \\
\boldsymbol{a} &: \mathit{string} \\
\boldsymbol{eat} &: \mathit{string} \\
\boldsymbol{to} &: \mathit{string} \\
\boldsymbol{wants} &: \mathit{string}
\end{aligned}$$

where, as usual, *string* is defined to be $o \to o$ for some atomic type o.

One then defines a morphism $(\mathcal{L}_{\text{SYNT}} : \Sigma_{\text{ABS}} \to \Sigma_{\text{S-FORM}})$ as follows:

$(\mathcal{L}_{\text{SYNT}})$

$$\begin{aligned}
NP &:= \mathit{string} \\
N &:= \mathit{string} \\
S &:= \mathit{string} \\
S_{\mathit{inf}} &:= \mathit{string} \\
VP &:= \mathit{string}
\end{aligned}$$

$$\begin{aligned}
\textsc{suzy} &:= \boldsymbol{Suzy} \\
\textsc{bill} &:= \boldsymbol{Bill} \\
\textsc{mushroom} &:= \boldsymbol{mushroom} \\
\textsc{a} &:= \lambda xy.\, y\,(\boldsymbol{a} + x) \\
\textsc{a}_{\mathit{inf}} &:= \lambda xy.\, y\,(\boldsymbol{a} + x) \\
\textsc{eat} &:= \lambda xy.\, y + \boldsymbol{eat} + x \\
\textsc{to} &:= \lambda x.\, \boldsymbol{to} + (x\,\epsilon) \\
\textsc{want} &:= \lambda xy.\, y + \boldsymbol{wants} + x
\end{aligned}$$

where, as usual, the concatenation operator $(+)$ is defined as functional composition, and the empty word (ϵ) as the identity function.

[1] 1. Give two different terms, say t_0 and t_1, such that:

$$\mathcal{L}_{\text{SYNT}}(t_0) = \mathcal{L}_{\text{SYNT}}(t_1) = \boldsymbol{Bill} + \boldsymbol{wants} + \boldsymbol{to} + \boldsymbol{eat} + \boldsymbol{a} + \boldsymbol{mushroom}$$

Exercise 6. One considers a third signature :

$$(\Sigma_{\text{L-FORM}}) \qquad \begin{aligned} \mathbf{suzy} &: \mathsf{ind} \\ \mathbf{bill} &: \mathsf{ind} \\ \mathbf{mushroom} &: \mathsf{ind} \to \mathsf{prop} \\ \mathbf{eat} &: \mathsf{ind} \to \mathsf{ind} \to \mathsf{prop} \\ \mathbf{want} &: \mathsf{ind} \to \mathsf{prop} \to \mathsf{prop} \end{aligned}$$

One then defines a morphism ($\mathcal{L}_{\text{SEM}} : \Sigma_{\text{ABS}} \to \Sigma_{\text{L-FORM}}$) as follows:

$$(\mathcal{L}_{\text{SEM}}) \qquad \begin{aligned} NP &:= \mathsf{ind} \\ N &:= \mathsf{ind} \to \mathsf{prop} \\ S &:= \mathsf{prop} \\ S_{inf} &:= \mathsf{prop} \\ VP &:= \mathsf{ind} \to \mathsf{prop} \\ \textsc{suzy} &:= \mathbf{suzy} \\ \textsc{bill} &:= \mathbf{bill} \\ \textsc{mushroom} &:= \mathbf{mushroom} \\ \textsc{a} &:= \lambda xy.\, \exists z.\, (x\, z) \wedge (y\, z) \\ \textsc{a}_{inf} &:= \lambda xy.\, \exists z.\, (x\, z) \wedge (y\, z) \\ \textsc{eat} &:= \lambda xy.\, \mathbf{eat}\, y\, x \\ \textsc{to} &:= \lambda x.\, x \\ \textsc{want} &:= \lambda xy.\, \mathbf{want}\, y\, (x\, y) \end{aligned}$$

[1] 1. Compute the different semantic interpretations of the sentence *Bill wants to eat a mushroom*, i.e., compute $\mathcal{L}_{\text{SEM}}(t_0)$ and $\mathcal{L}_{\text{SEM}}(t_1)$.

Exercise 7. One extends Σ_{ABS} and $\mathcal{L}_{\text{SYNT}}$, respectively, as follows:

$$(\Sigma_{\text{ABS}}) \quad \textsc{want2} : \quad VP \to NP \to S$$
$$(\mathcal{L}_{\text{SYNT}}) \quad \textsc{want2} := \lambda xyz.\, z + \boldsymbol{wants} + x + y$$

[1] 1. Extend $\mathcal{L}_{\text{SEM}}$ accordingly in order to allow for the analysis of a sentence such as *Bill wants Suzy to eat a mushroom.*

Exercise 8. One extends Σ_{ABS} as follows:

$$(\Sigma_{\text{ABS}}) \qquad \begin{aligned} \textsc{everyone} &: (NP \to S) \to S \\ \textsc{think} &: S \to NP \to S \end{aligned}$$

in order to allow for the analysis of the following sentence:

(1) *everyone thinks Bill wants to eat a mushroom.*

[3] 1. Extend $\Sigma_{\text{S-FORM}}$, $\mathcal{L}_{\text{SYNT}}$, $\Sigma_{\text{L-FORM}}$, and $\mathcal{L}_{\text{SEM}}$ accordingly.

[2] 2. Give the several λ-terms that correspond to the different parsings of sentence (1).