MPRI 2-27-1 Exam

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

1 Multiple Context-Free Grammars

Multiple Context-Free Grammars are a mildly context-sensitive formalism defined by Seki, Matsumura, Fuji, and Kasami in 1991. The purpose of this section is to instantiate the 'parsing as intersection' framework in their case.

Exercise 1 (Multiple Context-Free Grammars)**.** Let $\mathcal{X}$ be an infinite countable set of variables. A *multiple context-free grammar* (MCFG) of rank m and degree d is a tuple $\mathcal{G} = \langle N, \Sigma, P, S\rangle$ where N is a finite alphabet of nonterminals $A^{(r)}$ with positive ranks $0 < r \leq m$, Σ a finite alphabet of terminals, $S^{(1)} \in N$ is the start symbol with rank 1, and P is a finite set of productions p of form

$$A^{(r_0)}(\alpha_1, \ldots, \alpha_{r_0}) \coloneq B_1^{(r_1)}(x_{1,1}, \ldots, x_{1,r_1}), \ldots, B_k^{(r_k)}(x_{k,1}, \ldots, x_{k,r_k}). \tag{p}$$

where $0 \leq k \leq d$ is the degree of the production, $A^{(r_0)}, B_1^{(r_1)}, \ldots, B_k^{(r_k)}$ are nonterminals in N, $x_{1,1}, \ldots, x_{k,r_k}$ are distinct variables from $\mathcal{X}$, and $\alpha_1, \ldots, \alpha_{r_0}$ are *linear strings* α over $\Sigma \uplus \{x_{1,1}, \ldots, x_{k,r_k}\}$, i.e. strings where each variable $x_{i,j}$ occurs exactly once. Note that if $k = 0$, this entails $\alpha_j \in \Sigma^*$ for all $1 \leq j \leq r_0$.

An MCFG $\mathcal{G}$ defines a deduction system over *judgements* of form $\vdash_{\mathcal{G}} A^{(r)}(w_1, \ldots, w_r)$ where $A^{(r)}$ is a nonterminal from N and $w_1, \ldots, w_r$ are finite strings over Σ. This deduction system has a rule

$$\frac{\vdash_{\mathcal{G}} B_1^{(r_1)}(w_{1,1}, \ldots, w_{1,r_1}) \quad \cdots \quad \vdash_{\mathcal{G}} B_k^{(r_k)}(w_{k,1}, \ldots, w_{k,r_k})}{\vdash_{\mathcal{G}} A^{(r_0)}(\alpha_1\sigma, \ldots, \alpha_{r_0}\sigma)} \tag{$\vdash_{\mathcal{G}}$}$$

for every p in P and strings $w_{1,1}, \ldots, w_{k,r_k}$ in Σ^*, where σ is the substitution $x_{i,j} \mapsto w_{i,j}$.

The *language* of a nonterminal $A^{(r)}$ is the set of r-tuples of strings defined by

$$L_{\mathcal{G}}(A) \stackrel{\text{def}}{=} \{(w_1, \ldots, w_r) \in (\Sigma^*)^r \mid \vdash_{\mathcal{G}} A^{(r)}(w_1, \ldots, w_r)\} .$$

The *language* generated by $\mathcal{G}$ is accordingly the language of its start symbol $S^{(1)}$:

$$L(\mathcal{G}) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \vdash_{\mathcal{G}} S^{(1)}(w)\} .$$

Example 1 (Copy Language)**.** The MCFG $\mathcal{G}_{\text{copy}} = \langle \{S^{(1)}, A^{(2)}\}, \{a, b\}, P, S\rangle$ with productions

$$\begin{aligned} S^{(1)}(xy) &:\!- A^{(2)}(x, y).\\ A^{(2)}(ax, ay) &:\!- A^{(2)}(x, y).\\ A^{(2)}(bx, by) &:\!- A^{(2)}(x, y).\\ A^{(2)}(\varepsilon, \varepsilon) &:\!- . \end{aligned}$$

generates the language $L(\mathcal{G}_{\text{copy}}) = \{ww \mid w \in \{a, b\}^*\}$.

[1] 1. Give a MCFG for the language $L_{\text{cross}} \stackrel{\text{def}}{=} \{a^n b^m c^n d^m \mid n, m \geq 0\}$.

Solution: This language of cross-serial dependencies—which should remind you of the Swiss-German example of Shieber—is generated by the productions

$$\begin{aligned} S^{(1)}(x, y) &:\!- A^{(2)}(x, y).\\ A^{(2)}(ax, cy) &:\!- A^{(2)}(x, y).\\ A^{(2)}(xb, yd) &:\!- A^{(2)}(x, y).\\ A^{(2)}(\varepsilon, \varepsilon) &:\!- . \end{aligned}$$

[2] 2. Show that any context-free language is generated by an MCFG of rank 1.

Solution: Given a CFG $\mathcal{G} = \langle N, \Sigma, P, S\rangle$, we define the MCFG $\mathcal{G}' = \langle N, \Sigma, P', S\rangle$ with a production

$$A^{(1)}(u_0 x_1 u_1 \cdots u_{k-1} x_k u_k) :\!- B_1^{(1)}(x_1), \ldots, B_k^{(1)}(x_k).$$

in P' for each production $A \to u_0 B_1 u_1 \cdots u_{k-1} B_k u_k$ in P, where $u_0, \ldots, u_k$ are strings in Σ^* and $B_1, \ldots, B_k$ are nonterminals in N.

Exercise 2 (Emptiness of MCFGs)**.** The first main ingredient in the 'parsing as intersection' framework is to prove that the emptiness problem is decidable for MCFGs. In order to consider complexity questions, we define the size of a MCFG $\mathcal{G} = \langle N, \Sigma, P, S\rangle$ by summing $k + \sum_{j=1}^{r_0}(|\alpha_j| + 1)$ over all the productions p in P.

[3] 1. Show that there exists a linear-time algorithm that inputs an MCFG $\mathcal{G}$ and returns whether $L(\mathcal{G}) = \emptyset$.

Solution: We reduce in linear time the emptiness problem for MCFGs to that for CFGs, the latter problem being decidable in linear time (it is a variant of HornSAT).

The key argument is that, according to $(\vdash_{\mathcal{G}})$, $\vdash_{\mathcal{G}} A^{(r_0)}(w_{0,1}, \ldots, w_{0,r_0})$ holds for some strings $w_{0,1}, \ldots, w_{0,r_0}$ in Σ^* if and only if there exists a production p in P and the $1 \leq i \leq k$ judgements $\vdash_{\mathcal{G}} B^{(r_i)}(w_{i,1}, \ldots, w_{i,r_i})$ hold for some strings $w_{i,1}, \ldots, w_{i,r_i}$ in Σ^*.

We construct accordingly the CFG $\mathcal{G}' = \langle N, \Sigma, P', S\rangle$ with a production $A \to B_1 \cdots B_k$ for each production p in P. Then by the previous observation, $L_{\mathcal{G}}(A) = \emptyset$ iff $L_{\mathcal{G}'}(A) = \emptyset$ for all nonterminals A. This is clearly a linear-time reduction.

Exercise 3 (Intersection with a Regular Language)**.** The second ingredient of the 'parsing as intersection' framework is to show that the class of languages generated by MCFGs is closed under intersection with regular languages.

[2] 1. As a preliminary, show that for any MCFG, one can construct in linear time an equivalent MCFG where the productions p in P with $k > 0$ enforce $\alpha_j \in \mathcal{X}^*$, i.e. no terminal symbol appears in such productions, and each α_j is of form $y_1 \cdots y_{n_j}$ for $y_1, \ldots, y_{n_j}$ variables taken among $x_{1,1}, \ldots, x_{k,r_k}$.

Solution: Consider some production p in P with $k > 0$ and some $\alpha_j = u_{j,0}y_1u_{j,1}\cdots u_{j,n_j-1}y_{n_j}u_{j,n_j}$ with $y_1, \ldots, y_{n_j}$ variables among $x_{1,1}, \ldots, x_{k,r_k}$ and $u_{j,0}, \ldots, u_{j,n_j}$ strings in Σ^*. We introduce a fresh nonterminal $C^{(1)}_{j,\ell}$ and variable $z_{j,\ell}$ for each such $u_{j,\ell}$ and define $\alpha'_j \stackrel{\text{def}}{=} z_{j,0}y_1z_{j,1}\cdots z_{j,n_j-1}y_{n_j}z_{j,n_j}$ for each such α_j. We then replace p with

$$A^{r_0}(\alpha'_1, \ldots, \alpha'_{r_0}) \coloneq B^{(r_1)}_1(x_{1,1}, \ldots, x_{1,r_1}), \ldots, B^{(r_k)}_k(x_{k,1}, \ldots, x_{k,r_k}), C^{(1)}_{1,0}(z_{1,0}), \ldots, C^{(1)}_{r_0,n_{r_0}}(z_{r_0,n_{r_0}}).$$

and for all $1 \leq j \leq r_0$ and $0 \leq \ell \leq n_j$

$$C^{(1)}_{j,\ell}(u_{j,\ell}) \coloneq .$$

This transformation results in an increase in size by $\leq \sum_{j=1}^{r_0}(|\alpha_j| + 1) + k$ for each production p in P, hence a linear increase overall.

[5] 2. Show that, given an MCFG $\mathcal{G} = \langle N, \Sigma, P, S\rangle$ and a nondeterministic finite automaton $\mathcal{A} = \langle Q, \Sigma, \delta, I, F\rangle$, we can compute an MCFG $\mathcal{G}'$ such that $L(\mathcal{G}') = L(\mathcal{G}) \cap L(\mathcal{A})$ and $|\mathcal{G}'| \in O(|\mathcal{G}| \cdot |Q|^{m\max(k+1,2)})$.

Hint: Use nonterminals $A^{(r)}_{q_1,q'_1,\ldots,q_r,q'_r}$ such that $\vdash_{\mathcal{G}'} A^{(r)}_{q_1,q'_1,\ldots,q_r,q'_r}(w_1, \ldots, w_r)$ if and only if $\vdash_{\mathcal{G}} A^{(r)}(w_1, \ldots, w_r)$ and $q_j \xrightarrow{w_j}_{\mathcal{A}} q'_j$ for all $1 \leq j \leq r$.

Solution: We can assume without loss of generality that $\mathcal{G}$ is in the form of the previous question. We define two types of productions in P':

1. for each production of form
$$A^{(r_0)}(u_1, \ldots, u_{r_0}) \coloneq .$$
in P, we create a production in P'
$$A^{(r_0)}_{q_1,q'_1,\ldots,q_{r_0},q'_{r_0}}(u_0, \ldots, u_{r_0}) \coloneq .$$
for every $q_1, q'_1, \ldots, q_{r_0}, q'_{r_0}$ in Q such that $q_j \xrightarrow{u_j}_{\mathcal{A}} q'_j$ for each $1 \leq j \leq r_0$. This requires to quantify over $2r_0 \leq 2m$ states for each production in P with $k = 0$.

2. for each production of form (p) with $k > 0$ and each α_j of form $y_{j,1} \cdots y_{j,n_j}$, we create a production in P'
$$A^{(r_0)}_{q_1,q'_1,\ldots,q_{r_0},q'_{r_0}}(\alpha_1, \ldots, \alpha_{r_0}) \coloneq B^{(r_1)}_{1,q_{1,1},q'_{1,1},\ldots,q_{1,r_1},q'_{1,r_1}}(x_{1,1}, \ldots, x_{1,r_1}), \ldots, B^{(r_k)}_{k,q_{1,k},q'_{1,k},\ldots,q_{k,r_k},q'_{k,r_k}}(x_{k,1}, \ldots, x_{k,r_k}).$$
for all choices of states $q_0, q'_0, \ldots, q_{r_0}, q'_{r_0}, q_{1,1}, \ldots, q'_{k,r_k}$ such that, for every $1 \leq j \leq r_0$,
   - if $y_{j,1} = x_{i,\ell}$ for some $1 \leq i \leq k$ and $1 \leq \ell \leq r_i$, then $q_j = q_{i,\ell}$,
   - if $y_{j,n_j} = x_{i,\ell}$ for some $1 \leq i \leq k$ and $1 \leq \ell \leq r_i$, then $q'_j = q'_{i,\ell}$, and
   - for every $1 \leq j < n_j - 1$, if $y_{j,j+1} = x_{i,\ell}$ and $y_{j+1,j+2} = x_{i',\ell'}$ for some $1 \leq i, i' \leq k$, $1 \leq \ell \leq r_i$, and $q \leq \ell' \leq r_{i'}$, then $q'_{i,\ell} = q_{i',\ell'}$.

   We need to pick $2\sum_{i=0}^{k} r_i$ states for each such p in P, but among those $2r_0$ are equal to some states in $\{q_{1,1}, \ldots, q'_{k,r_k}\}$, and exactly half of the remaining $2\sum_{i=1}^{k} r_i - 2r_0$ states are unconstrained. Hence, we only need to quantify over $\sum_{i=0}^{k} r_i \leq (k+1)m$ states for each p with $k > 0$.

[1] 3. Deduce an algorithm for the *membership problem*, which given an MCFG $\mathcal{G} = \langle N, \Sigma, P, S \rangle$ and a string w in Σ^*, returns whether $w \in L(\mathcal{G})$.

Solution: As usual in the 'parsing as intersection' framework, given w, build $\mathcal{A}_w$ with language $\{w\}$ and $|w|$ states, then use the previous question to construct $\mathcal{G}'$ for the language $L(\mathcal{G}') = L(\mathcal{G}) \cap \{w\}$ and the previous exercise to test whether $L(\mathcal{G}') = \emptyset$, which occurs if and only if $w \notin L(\mathcal{G})$.

2 Covert Movements in Second-Order ACGs

In the exercises that follow, one only considers 2nd-order ACGs. This allows one not to bother about linearity constraints, and to work in the setting of the simply-typed λ-calculus.

Exercise 4. One considers the three following signatures:

$$(\Sigma_{\text{ABS}}) \qquad \begin{aligned} \textsc{pierre} &: NP \\ \textsc{maison} &: N \\ \textsc{une} &: N \to QNP \\ \textsc{acheter} &: QNP \to VP \\ \textsc{veux} &: VP \to NP \to S \end{aligned}$$

$$(\Sigma_{\text{S-FORM}}) \qquad \begin{aligned} /Pierre/ &: string \\ /maison/ &: string \\ /une/ &: string \\ /acheter/ &: string \\ /veux/ &: string \end{aligned}$$

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

$$(\Sigma_{\text{L-FORM}}) \qquad \begin{aligned} \mathbf{p} &: \mathsf{ind} \\ \mathbf{house} &: \mathsf{ind} \to \mathsf{prop} \\ \mathbf{buy} &: \mathsf{ind} \to \mathsf{ind} \to \mathsf{prop} \\ \mathbf{want} &: \mathsf{ind} \to \mathsf{prop} \to \mathsf{prop} \end{aligned}$$

One then defines two morphisms ($\mathcal{L}_{\text{SYNT}} : \Sigma_{\text{ABS}} \to \Sigma_{\text{S-FORM}}$, and $\mathcal{L}_{\text{SEM}} : \Sigma_{\text{ABS}} \to \Sigma_{\text{L-FORM}}$) as follows:

$$(\mathcal{L}_{\text{SYNT}}) \qquad \begin{aligned} NP &:= string \\ N &:= string \\ QNP &:= string \\ VP &:= string \\ S &:= string \\ \\ \textsc{pierre} &:= /Pierre/ \\ \textsc{maison} &:= /maison/ \\ \textsc{une} &:= \lambda x.\, /une/ + x \\ \textsc{acheter} &:= \lambda x.\, /acheter/ + x \\ \textsc{veux} &:= \lambda xy.\, y + /veux/ + x \end{aligned}$$

where, as usual, the concatenation operator ($+$) is defined as functional composition.

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

$$\begin{aligned} NP &:= (\mathsf{ind} \to \mathsf{prop}) \to \mathsf{prop} \\ N &:= \mathsf{ind} \to \mathsf{prop} \\ QNP &:= (\mathsf{ind} \to \mathsf{prop}) \to \mathsf{prop} \\ VP &:= \mathsf{ind} \to \mathsf{prop} \\ S &:= \mathsf{prop} \end{aligned}$$

$$\begin{aligned} \textsc{Pierre} &:= \lambda x.\, x\,\mathbf{p} \\ \textsc{maison} &:= \mathbf{house} \\ \textsc{une} &:= \lambda xy.\, \exists z.\, (x\,z) \wedge (y\,z) \\ \textsc{acheter} &:= \lambda xy.\, x\,(\lambda z.\, \mathbf{buy}\,y\,z) \\ \textsc{veux} &:= \lambda xy.\, y\,(\lambda z.\, \mathbf{want}\,z\,(x\,z)) \end{aligned}$$

[2] 1. Check that $\mathcal{L}_{\text{SEM}}$ is such that the interpretation it gives to ACHETER is consistent with the interpretation it gives to the types.

Solution: We have that ACHETER is of type $QNP \to VP$. Then, we have that :

$$\mathcal{L}_{\text{SEM}}(QNP \to VP) = ((\mathsf{ind} \to \mathsf{prop}) \to \mathsf{prop}) \to \mathsf{ind} \to \mathsf{prop} \tag{1}$$

We then compute the principal typing of the interpretation of ACHETER:

$$\lambda xy.\, x\,(\lambda z.\, \mathbf{buy}\,y\,z) : ((\mathsf{ind} \to \mathsf{prop}) \to \alpha) \to \mathsf{ind} \to \alpha \tag{2}$$

Finally, we check that (1) is an instance of (2).

[1] 2. Give a term, say t, such that:

$$\mathcal{L}_{\text{SYNT}}(t) = /Pierre/ + /veux/ + /acheter/ + /une/ + /maison/$$

Then, compute $\mathcal{L}_{\text{SEM}}(t)$.

Solution:

$$t = \textsc{veux}\,(\textsc{acheter}\,(\textsc{une}\,\textsc{maison}))\,\textsc{pierre}$$

$$\mathcal{L}_{\text{SEM}}(t) = \mathbf{want}\,\mathbf{p}\,(\exists z.\, (\mathbf{house}\,z) \wedge (\mathbf{buy}\,\mathbf{p}\,z))$$

Exercise 5. One extends Σ_{ABS} with the following constants (and types):

$$\begin{aligned} \textsc{trace} &: XNP \\ \textsc{x-acheter} &: XNP \to XVP \\ \textsc{x-veux} &: XVP \to NP \to XS \\ \textsc{qr} &: QNP \to XS \to S \end{aligned}$$

Accordingly, one extends $\mathcal{L}_{\text{SYNT}}$ as follows:

$$\begin{aligned} XNP &:= string \to string \\ XVP &:= string \to string \\ XS &:= string \to string \end{aligned}$$

$$\begin{aligned} \textsc{trace} &:= \lambda x.\, x \\ \textsc{x-acheter} &:= \lambda xy.\, /acheter/ + (x\, y) \\ \textsc{x-veux} &:= \lambda xyz.\, y + /veux/ + (x\, z) \\ \textsc{qr} &:= \lambda xy.\, y\, x \end{aligned}$$

[1] 1. Compute the interpretation of the following term (according to the above extension of $\mathcal{L}_{\text{SYNT}}$):

$$\textsc{qr}\,(\textsc{une maison})\,(\textsc{x-veux}\,(\textsc{x-acheter trace})\,\textsc{pierre}) \qquad (t_{\text{re}})$$

Solution:

$$\mathcal{L}_{\text{SYNT}}(t_{\text{re}}) = /Pierre/ + /veux/ + /acheter/ + /une/ + /maison/$$

Exercise 6. One also extends $\mathcal{L}_{\text{SEM}}$ as follows:

$$\begin{aligned} XNP &:= \mathsf{ind} \to (\mathsf{ind} \to \mathsf{prop}) \to \mathsf{prop} \\ XVP &:= \mathsf{ind} \to \mathsf{ind} \to \mathsf{prop} \\ XS &:= \mathsf{ind} \to \mathsf{prop} \end{aligned}$$

$$\begin{aligned} \textsc{trace} &:= \lambda xy.\, y\, x \\ \textsc{x-acheter} &:= \lambda wxy.\, w\, x\, (\lambda z.\, \mathbf{buy}\, y\, z) \\ \textsc{x-veux} &:= \cdots \\ \textsc{qr} &:= \cdots \end{aligned}$$

[3] 1. Complete the above extension (i.e., provide the interpretations of X-VEUX and QR) in such a way that $\mathcal{L}_{\text{SEM}}(t_{\text{re}})$ yields a *de re* interpretation.

Solution: Let:

$$\begin{aligned} \textsc{x-veux} &:= \lambda wxy.\, x\, (\lambda z.\, \mathbf{want}\, z\, (w\, y\, z)) \\ \textsc{qr} &:= \lambda xy.\, x\, y \end{aligned}$$

Then:

$$\mathcal{L}_{\text{SEM}}(t_{\text{re}}) = \exists z.\, (\mathbf{house}\, z) \wedge (\mathbf{want}\, \mathbf{p}\, (\mathbf{buy}\, \mathbf{p}\, z))$$