Compiler Theory CS447a: Quiz 2. **UWO, October 21, 2004.**

Student ID number:

Guidelines. The quiz is closed book and all notes are forbidden. The duration of the quiz is 40 minutes. Each of the 4 exercices will marked be over 5 points. All answers should be written in the *answer boxes*. No justifications for the answers are needed **unless** explicitly required. You are expected to do this exam on your own without assistance from anyone else in the class. If possible, please avoid pencils and use pens with dark ink. Thank you.

Exercise 1. We consider the alphabet $\Sigma = \{(,)\}$ consisting of the opening parenthesis and the closing parenthesis. For each of the following grammars, if its generated language is regular, then give a finite automaton recognizing this language, otherwise justify briefly why it is not regular.

$$G_1 : \left\{ \begin{array}{lll} S & \longmapsto & A \\ A & \longmapsto & (\,B \\ A & \longmapsto & (\,A \\ B & \longmapsto &)\,B \\ B & \longmapsto &) \end{array} \right. \qquad G_2 : \left\{ \begin{array}{lll} S & \longmapsto & E \\ E & \longmapsto & (\,E\,)\,E \\ E & \longmapsto & \varepsilon \end{array} \right.$$

Answer 1

Exercise 2. We consider the alphabet $\Sigma = \{a, b\}$ and the grammar G below. Let L be the language generated by G over Σ.

$$G \left\{ \begin{array}{lcl} S & \longmapsto & AB \\ A & \longmapsto & aAA \\ A & \longmapsto & \varepsilon \\ B & \longmapsto & bBB \\ B & \longmapsto & \varepsilon \end{array} \right.$$

Give a grammar G' generating $L \setminus \{\varepsilon\}$ and with **no** ε-productions, that is **no** rules of the form $X \longmapsto \varepsilon$, where ε denotes the empty word. Is G' left-recursive?

Answer 2

Exercise 3. Consider the following grammar G with terminals a, b, c, d and nonterminals S, A, B, C.

$$\begin{array}{lcl} S & \longmapsto & C\,A \\ A & \longmapsto & a\,B\,A \mid \varepsilon \\ B & \longmapsto & b\,B \mid C\,d \\ C & \longmapsto & c\,C \mid \varepsilon \end{array}$$

We recall the rules for the computation of the FOLLOW set of each nonterminal.

$$\begin{array}{llll} \text{if} & A \longmapsto \alpha B \beta & \text{then} & \text{FOLLOW}(B) := \text{FIRST}(\beta) \setminus \{\varepsilon\} \cup \text{FOLLOW}(B) \\ \text{if} & A \longmapsto \alpha B & \text{then} & \text{FOLLOW}(B) := \text{FOLLOW}(B) \cup \text{FOLLOW}(A) \\ \text{if} & \begin{cases} A \longmapsto \alpha B \beta \\ \beta \stackrel{*}{\Longrightarrow} \varepsilon \end{cases} & \text{then} & \text{FOLLOW}(B) := \text{FOLLOW}(B) \cup \text{FOLLOW}(A) \end{array}$$

Recall also that if \$ denotes the end-of-line symbol and if S is the start symbol of the grammar (as it is the case above) then we have $\$ \in \text{FOLLOW}(S)$. Give the parsing table of G (for the non-recursive implementation of predictive parsing).

Answer 3

Annexe 1. Recall that for any string α of symbols the set FIRST(α) satisfies the following conditions for every terminal a and every string of symbols β

- FIRST(α) $\subseteq V_T \cup \{\varepsilon\}$
- $\alpha \stackrel{*}{\Longrightarrow} a\beta$ iff $a \in$ FIRST(α)
- $\alpha \stackrel{*}{\Longrightarrow} \varepsilon$ iff $\varepsilon \in$ FIRST(α).

For a symbol $X \in V_T \cup V_N$ the set FIRST(X) can be computed as follows

Algorithm 1

Input: $X \in V_T \cup V_N$ where V_T and V_N are the sets of terminals and nonterminals, respectively.

Output: FIRST(X).

```
if X ∈ V_T then
      FIRST(X) := {X}
else
      if X ⟼ ε then
         FIRST(X) := {ε}
      else
         FIRST(X) := ∅
for every X ⟶ α with α ≠ ε repeat
      let α = X_1 X_2 ··· X_ℓ where each X_i is a grammar symbol
      for i = 1 ··· ℓ repeat
            FIRST(X) := (FIRST(X_i) \ {ε}) ∪ FIRST(X)
      until ε ∉ FIRST(X_i)
      if ε ∈ ⋂_{1≤i≤ℓ} FIRST(X_i) then
            FIRST(X) := FIRST(X) ∪ {ε}
```

Algorithm 2

Input: $X = X_1 X_2 \cdots X_n$ where every $X_i \in V_T \cup V_N$.

Output: FIRST($X_1 X_2 \cdots X_n$).

FIRST(X) := $\emptyset$
for $i = 1 \cdots n$ **repeat**
  FIRST(X) := (FIRST(X_i) $\setminus \{\varepsilon\}$) $\cup$ FIRST(X)
until $\varepsilon \notin$ FIRST(X_i)
if $\varepsilon \in \bigcap_{1 \leq i \leq n}$FIRST($X_i$) **then**
  FIRST(X) := FIRST(X) $\cup \{\varepsilon\}$

Recall that FOLLOW(A) is the set of the terminals that can appear immediately to the right of the nonterminal A in some sentential form. Moreover \$ belongs to FOLLOW(A) if A is the rightmost symbol in some sentential form.

Algorithm 3

Input: $G = (V_T, V_N, S, P)$.

Output: Parsing table M.

for every $A \in V_N$ **repeat**
  for every $a \in V_T \cup \{\$\}$ **repeat**
    $M[A, a]$:= $\emptyset$
for every $(A \longmapsto \alpha) \in P$ **repeat**
  for every $a \in$ FIRST(α) **repeat**
    if $\alpha \neq \varepsilon$ **then**
      $M[A, a]$:= $M[A, a] \cup \{A \longmapsto \alpha\}$
  if $\varepsilon \in$ FIRST(α) **then**
    for every $b \in$ FOLLOW(A) **repeat**
      $M[A, b]$:= $M[A, b] \cup \{A \longmapsto \alpha\}$
for every $A \in V_N$ **repeat**
  for every $a \in V_T \cup \{\$\}$ **repeat**
    if $M[A, a] = \emptyset$ **then**
      $M[A, a]$:= *error*