40-414 – Compiler Design, Fall 1400, Sharif University of Technology

Writing Assignment II

Released: Sunday, 16/08/1400

Due: Friday, 28/08/1400 at 11:59pm

This assignment covers lecture notes 4, 5, and 6 on the topic of parsing. You can discuss and work out this assignment with other students. However, your write-up must be your own individual work. A PDF format of the solution **(with your full name and student number)** should be uploaded in the course page in Quera **before 11:59 PM, Friday 28/08/1400** (https://quera.ir/course/9134/). Submissions with more than 50 hours delay will not be graded. Submissions with less than 25 hours delay will be penalized by the following rule:

Penalized mark = M * (100 – 2 * D) / 100

Where M = the mark achieved from your solution and D is number of hours passed the deadline. Submissions with 25 < X ≤ 50 hours delay will be penalized by P.M. = M * 0.5.

Note: Throughout this assignment, Capital letters indicate non-terminals and ε is epsilon.

1. Give the context-free grammar (CFG) for the following language. Any grammar is acceptable - including ambiguous grammars - as long as it has the correct language.

- The set of all strings over {0, 1} that have even length and the first half of the strings is not equal to the second half.

2. (a) Left factor the following grammar:

```
S → S + S | S + P
P → P * P | P * I
I → − I | ( S ) | D
D → 0 | 1 N
N → 0 | 1 | N N | ε
```

(b) Eliminate left recursion from the following grammar:

```
S → S a S | U
U → U u U | T
T → t | f | T n | ( S )
```

3. Consider the grammar introduced here and answer the question (bold symbols are terminals):

- How many parse trees are possible for the string: "**a and a or a**"? Explain your answer (S is the start symbol).

```
S → S and S
  | S or S
  | T
  | a
T → a
```

4. Consider the following grammar over the alphabet {a, b, c}:

```
S → X a
X → b X
X → Y
Y → Z c
Z → b Z
Z → ε
```

(a) Prove that this grammar is not LL(1).

(b) It is possible to drop exactly one production from this grammar to obtain a new grammar generating the same language that is an LL(1) grammar. Identify this production and prove that the resulting grammar is LL(1).

5. Consider the following CFG, where the set of terminals is {0, 1, (,), ;}:

```
S → ( T
T → C A | )
A → ; B | )
B → C A | )
C → 0 | 1 | S
```

(a) Construct the FIRST sets for each of the non-terminals.

(b) Construct the FOLLOW sets for each of the non-terminals.

(c) Construct the LL(1) parsing table for the grammar. (Your parsing table must be sorted in each axis. i.e., Symbols must be sorted lexicographically from up to down and symbols must be sorted from left to right).

(d) Show the sequence of stack, input and action configurations that occur during an LL(1) parse of the string "(( ) ; 0)". At the beginning of the parse, the stack should contain a single S.

6. Convert the given grammar to a transition diagram and simplify it as much as possible:

```
        S' → S
1       S → A C
2-3     C → f | ε
4-5     A → a B C d | B Q
6-7     B → b B | ε
8-9     Q → q | ε
```

7. Consider the grammar given above (i.e., question 6).

(a) Draw the SLR(1) transition diagram and SLR(1) parsing table for this grammar. Note that $S' \rightarrow S$ has already been added.

(b) Draw the SLR(1) parsing table for this grammar.

(c) Show the sequence of stack, input, and action configurations that occur during an LR(1) parsing of string: **'a b b d f $'**.