Ambo University @ Woliso Campus

School of Technology and Informatics

Department of Computer Science

Compiler Design Assignment

for

3rd year Computer Science

Regular Students.

NOTE:

- Submission of the assignment before DEADLINE is possible.
- But after deadline assignment will be received with **deduction of mark.**
- Submission Date should be **June 25, 2020.**
- Copying from others makes your mark zero.
- Please do the assignment carefully and completely. You may evaluated by it only for a course.
- *Submit your assignment in .**pdf** format or by capturing photo and scanning as a single file if it is hand writing.*
- *Unclear, deleted, unreadable and insensible* answer makes your work **valueless**.
- *You can ask me any question, suggestion, unclear or doubt ideas about a course and assignment via one of bellow addresses.*
- Submit your assignment for me through my:
  - ***e-mail:*** **yoobsanb3@gmail.com**
  - ***telegram:*** **Yoobsan B Begi** ***or*** **0934407791**

Prepared and compiled by: ***Yoobsan Bechera***

May/26/2020, AU, Oromia, Ethiopia

1) Consider the context-free grammar S -> 55 + | 55 * | a
    a) Show how the string **aa+a*** can be generated by this grammar.
    b) Construct a parse tree for this string.
    c) What language does this grammar generate? Justify your answer.
2) Construct a syntax-directed translation scheme that translates arithmetic expressions from infix notation into prefix notation in which an operator appears before its operands; e.g., -xy is the prefix notation for x-y. Give annotated parse trees for the inputs 9-5+2 and 9-5*2.
3) Construct a syntax-directed translation scheme that t translates roman numerals into integers.
4) Construct a syntax-directed translation scheme that t translates postfix arithmetic expressions into equivalent infix arithmetic expressions.
5) Construct a syntax-directed translation scheme that translates arithmetic expressions from postfix notation into infix notation. Give annotated parse trees for the inputs 95-2* and 952*-
6) Construct a syntax-directed translation scheme that translates integers into roman numerals.
7) Construct recursive-descent parsers, starting with the following grammars:
    a) S -> + SS | -SS | a
    b) S -> 5(5) 5 | e
    c) S 0 5 1 | 0 1
8) Construct DFAs for the string matched by the following definition:
    a) digit =[0-9]
    b) nat=digit+
    c) signednat=(+|-)?nat
    d) number=signednat(“.”nat)?(E signedNat)?
9) Regular expression Consider the regular expression r = (a|b)*abb, that matches {abb, aabb, babb, aaabb, bbabb, ababb, aababb, ……}
    a) Construct a NFA from this, use Thompson’s construction.
    b) Construct a DFA from this NFA.
    c) Built a Transition Table.
10) Using the grammar below, construct a parse tree for the following string using RDP algorithm: ((**id . id**) **id** (**id**) (( )))

$S \rightarrow E$

$E \rightarrow id$

$\quad | \; (\; E \; . \; E \;)$

$\quad | \; (\; L \;)$

$\quad | \; (\;)$

$L \rightarrow L \; E$

$\quad | \; E$

11) Consider the following grammar over the alphabet { g,h,i,b}

A → BCD

B → bB | ε

C → Cg | g | Ch | i

D → AB | ε

a) Fill in the table below with the FIRST and FOLLOW sets for the non-terminals in this grammar:

12) Let G be the following grammar:

S → [SX] | a
X → ε | +SY | Yb
Y → ε | -SXc

a) Find FIRST and FOLLOW sets for the non-terminals in this grammar.
b) Construct predictive parsing table for the grammar above.
c) Show a top down parse of the string [a+a-ac]

13) Consider the following grammar:

S → ScB | B
B → e | efg | efCg
C → SdC | S

a) Justify whether the grammar is LL(1) or not?
b) If not, translate the grammar into LL(1).
c) Construct predictive parsing table for the above grammar.

14) *Given the following Grammar:*

S → A
S → B
A → a A b
A → 0
B → a B b b
B → 1

a) Construct the SLR parsing table.
b) Write the action of an LR parse for the following string **aa1bbbb**

GOOD LUCK!