CS3342 – Assignment 1
due Feb. 9, 2023
2-day no-penalty extension until: Feb. 11, 11:55pm

1. (10pt) Write a regular expression for comments in a Python program. If necessary, you can use the notation `not-a` to denote all characters different from `a`.

2. (15pt) A scanner is built for a language where the identifiers start with a letter followed by any number of letters or digits.

   (a) (5pt) Draw a DFA that accepts all identifiers and nothing else.

   (b) (10pt) Assume that a new rule is imposed, that all identifiers that contain digits must have odd length; everything else stays the same. Draw a DFA for identifiers under the new restrictions.

3. (25pt) Consider the following grammar, G, for conditional statements:

   1. $P \longrightarrow S$ `$$`
   2. $S \longrightarrow B$
   3. $S \longrightarrow U$
   4. $B \longrightarrow$ `if` C `then` B `else` B
   5. $B \longrightarrow O$
   6. $U \longrightarrow$ `if` C `then` S
   7. $U \longrightarrow$ `if` C `then` B `else` U
   8. $C \longrightarrow \mathtt{c}_i, i \geq 1$
   9. $O \longrightarrow \mathtt{s}_i, i \geq 1$

   Nonterminals: P, S, B, U, C, O; terminals: `if`, `then`, `else`, $\mathtt{c}_i$, $\mathtt{s}_i$, `$$`.

   (a) (2pt) Show the parse tree of G for the input:

   `if` $\mathtt{c}_1$ `then if` $\mathtt{c}_2$ `then` $\mathtt{s}_1$ `else if` $\mathtt{c}_3$ `then` $\mathtt{s}_2$ `$$` .

   (b) (3pt) Compute FIRST(X), FOLLOW(X), for all nonterminals X, and PREDICT(p), for all productions $p, 1 \leq p \leq 9$.

   (c) (5pt) Prove that G is not LL(1). Indicate all conflicts, that is, tokens belonging to two PREDICT(p) sets with the same LHS.

   (d) (10pt) Employ, on G, the techniques we used for attempting to make a grammar LL(1). Try to address all conflicts discovered at (c).

   (e) (5pt) Explain why it does not seem possible to obtain an LL(1) grammar.

4. (50pt) Write a Python program `comm_rm.py` to remove all comments from a C++ program. The program should work as follows:

   ```
   comm_rm inputC.cpp inputC_rm.cpp
   ```

   where `inputC.cpp` is any (correct) C++ program and `inputC_rm.cpp` is the same program with comments removed.

READ ME! Submit your answers as a *single pdf file* in OWL. Solutions should be typed; readable (by others!) hand-written solutions are also acceptable. Source code, if required, is submitted as separate files.

JFLAP: You are allowed to use JFLAP to help you solve the assignment. Make sure you understand what it does; JFLAP will not be available during in-person exams!

LaTeX: For those interested, the best (the only!) program for scientific writing is LaTeX. It is free and you can start using it in minutes: `https://tobi.oetiker.ch/lshort/lshort.pdf`

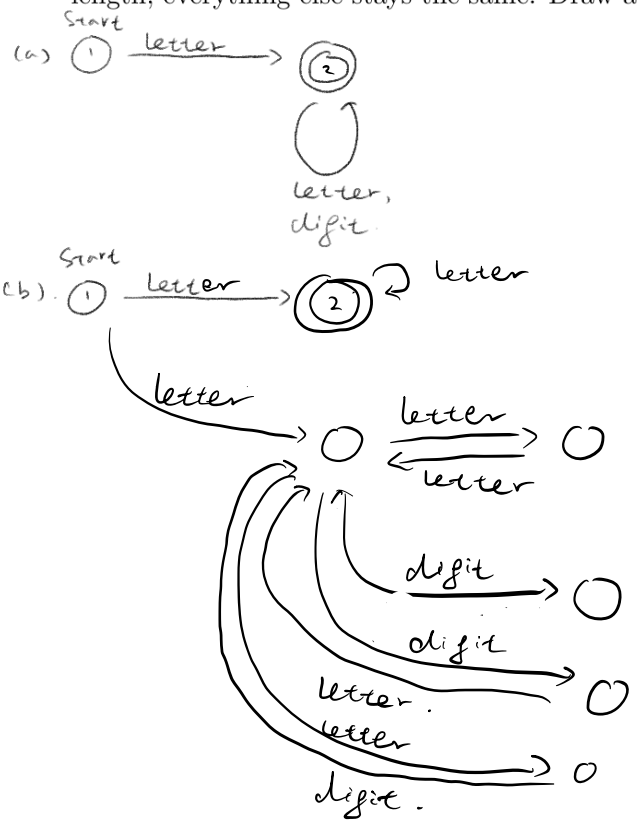
1. (10pt) Write a regular expression for comments in a Python program. If necessary, you can use the notation `not-a` to denote all characters different from `a`.

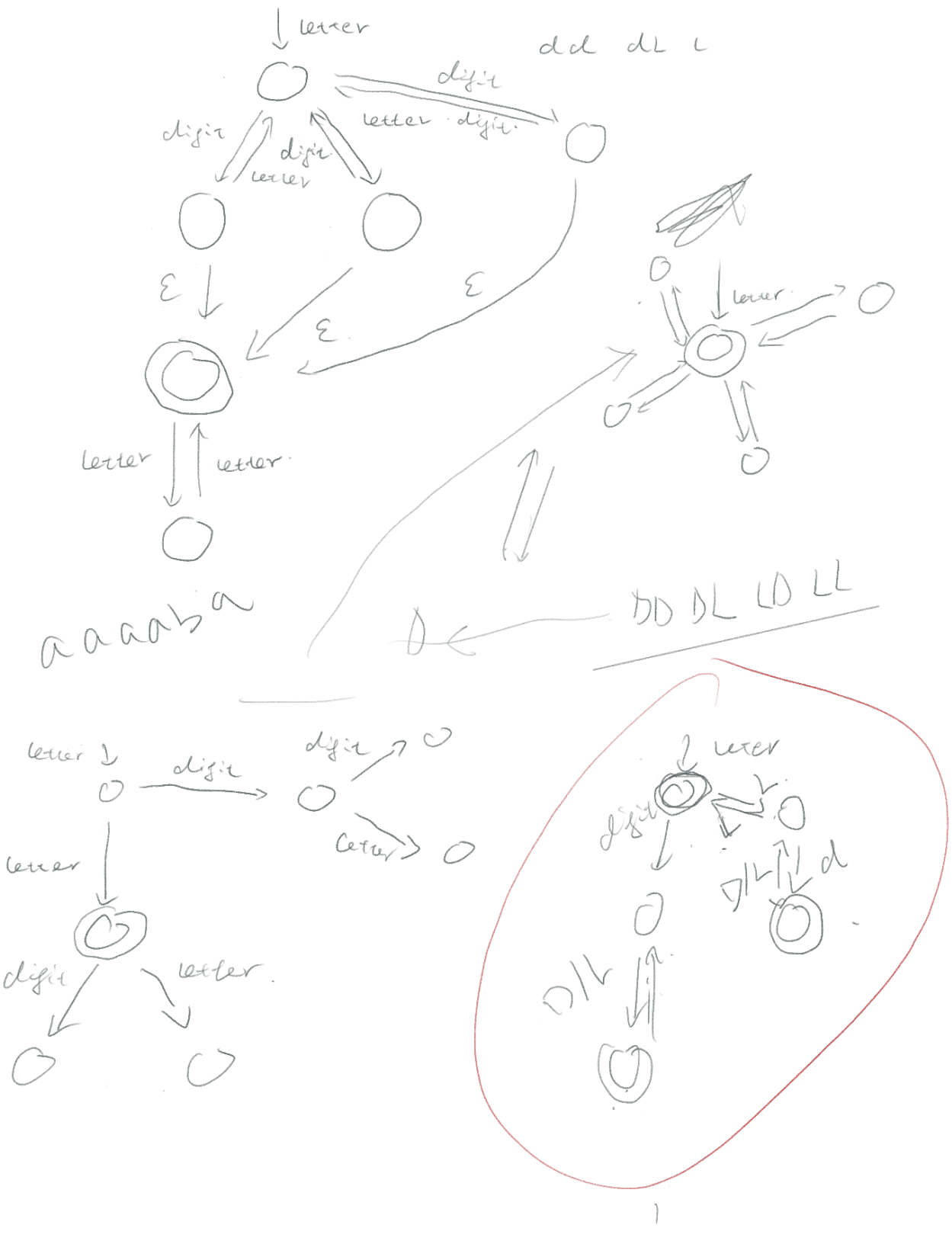
comment -> # {a, not-a}* | ''' {a, not-a}* ''' | """ {a, not-a}* """

2. (15pt) A scanner is built for a language where the identifiers start with a letter followed by any number of letters or digits.

   (a) (5pt) Draw a DFA that accepts all identifiers and nothing else.

   (b) (10pt) Assume that a new rule is imposed, that all identifiers that contain digits must have odd length; everything else stays the same. Draw a DFA for identifiers under the new restrictions.




letter
digit
digit
letter
digit
letter
digit
ε
ε
ε
letter
letter
letter
aaaaba
DD DL LD LL
letter
digit
digit
letter
letter
digit
letter
digit
letter
letter
d

3. (25pt) Consider the following grammar, G, for conditional statements:

1.	P	$\longrightarrow$	S `$$`	6.	U	$\longrightarrow$	`if` C `then` S
2.	S	$\longrightarrow$	B	7.	U	$\longrightarrow$	`if` C `then` B `else` U
3.	S	$\longrightarrow$	U	8.	C	$\longrightarrow$	$\texttt{c}_i, i \geq 1$
4.	B	$\longrightarrow$	`if` C `then` B `else` B	9.	O	$\longrightarrow$	$\texttt{s}_i, i \geq 1$
5.	B	$\longrightarrow$	O				

Nonterminals: P, S, B, U, C, O; terminals: `if`, `then`, `else`, $\texttt{c}_i$, $\texttt{s}_i$, `$$`.

(a) (2pt) Show the parse tree of G for the input:

$$\texttt{if c}_1 \texttt{ then if c}_2 \texttt{ then s}_1 \texttt{ else if c}_3 \texttt{ then s}_2 \texttt{ \$\$}\ .$$

(b) (3pt) Compute FIRST(X), FOLLOW(X), for all nonterminals X, and PREDICT(p), for all productions $p, 1 \leq p \leq 9$.

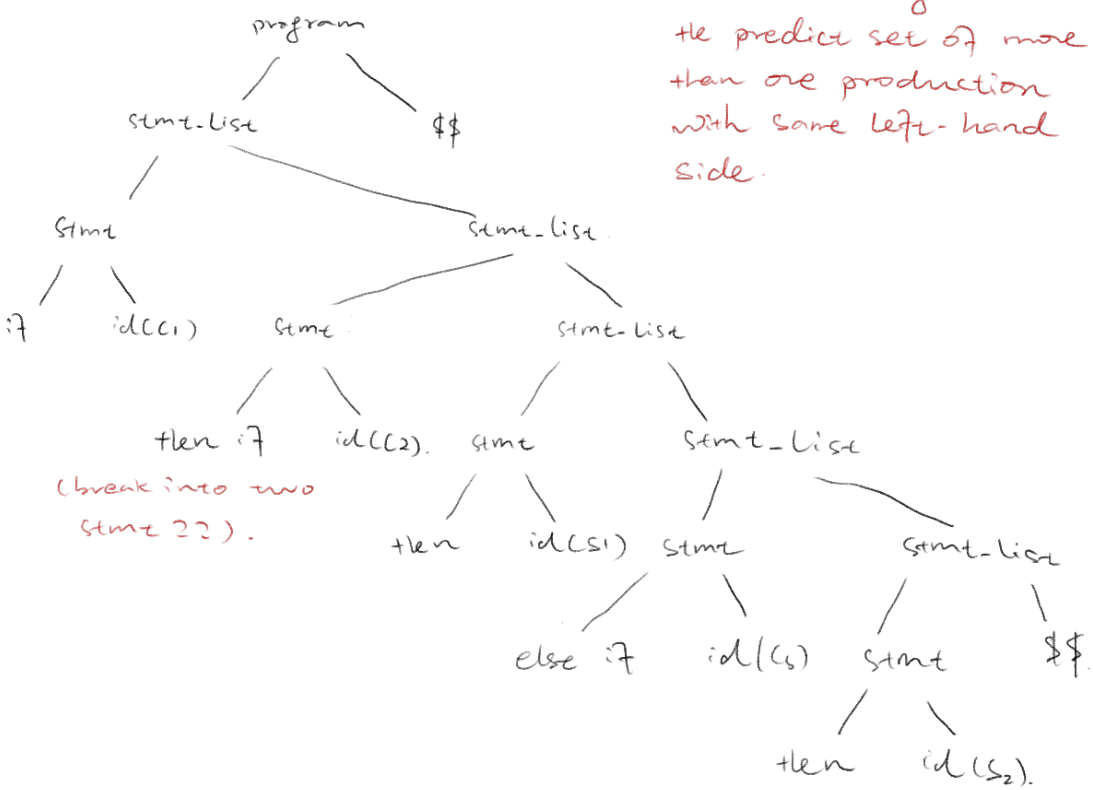
(c) (5pt) Prove that G is not LL(1). Indicate all conflicts, that is, tokens belonging to two PREDICT(p) sets with the same LHS.

(d) (10pt) Employ, on G, the techniques we used for attempting to make a grammar LL(1). Try to address all conflicts discovered at (c).

(e) (5pt) Explain why it does not seem possible to obtain an LL(1) grammar.

(e): a token belongs to the predict set of more than one production with same left-hand side.

(a)



(break into two stmt ??).

1. $P \longrightarrow S$ $$
2. $S \longrightarrow B$
3. $S \longrightarrow U$
4. $B \longrightarrow$ `if` C `then` B `else` B ?
5. $B \longrightarrow O$
6. $U \longrightarrow$ `if` C `then` S
7. $U \longrightarrow$ `if` C `then` B `else` U
8. $C \longrightarrow \mathtt{c}_i, i \geq 1$
9. $O \longrightarrow \mathtt{s}_i, i \geq 1$

Nonterminals: P, S, B, U, C, O; terminals: `if`, `then`, `else`, $\mathtt{c}_i$, $\mathtt{s}_i$, `$$`.

P -> S
-> B | U
-> ¬ C | O | ¬ C
-> ¬ c_i | s_i.

S -> ¬ c_i | s_i.
B -> ¬ c_i | s_i.
U -> ¬ c_i.
O -> s_i.

PREDICTION:

P -> S $$ {¬ c_i, s_i}.
S -> B {¬ c_i, s_i}
S -> U {¬ c_i}.] conflict?
B -> ¬ C then B else B {¬ c_i}
B -> O {s_i}.]

1. $P \longrightarrow S$ \$\$
2. $S \longrightarrow B$
3. $S \longrightarrow U$
4. $B \longrightarrow$ `if` C `then` B `else` B
5. $B \longrightarrow O$
6. $U \longrightarrow$ `if` C `then` S
7. $U \longrightarrow$ `if` C `then` B `else` U
8. $C \longrightarrow \mathtt{c}_i, i \geq 1$
9. $O \longrightarrow \mathtt{s}_i, i \geq 1$

Nonterminals: P, S, B, U, C, O; terminals: `if`, `then`, `else`, $\mathtt{c}_i$, $\mathtt{s}_i$, \$\$.

(b) (3pt) Compute FIRST(X), FOLLOW(X), for all nonterminals X, and PREDICT(p), for all productions $p, 1 \leq p \leq 9$.

N	P	S	B	U	C	O
FIRST(N)	{if, s_i}	{if, s_i}	if	if	c_i	s_i
	{if c_i, s_i}	{if c_i, s_i}	{if c_i, s_i}	if c_i		
FOLLOW(N)	{\$\$}	s_i	s_i	s_i	c_i	s_i
FOLLOW	ε	{\$\$}	{else}	ε	{then}	ε

PREDICT(N):

$P \to S$ \$\$ {if c_i, s_i}

$S \to B$ {s_i}

$S \to U$ {s_i}

$B \to$ if C then B else B {if}

$B \to O$ {s_i}

$O \to s_i$ {s_i}

$U \to$ if C then S {if c_i, s_i}

$U \to$ if C then B else U {if c_i, s_i}

$P \to S \to B \to O \to s_i$

U

O

FIRST

: => ε :f

(c) (5pt) Prove that G is not LL(1). Indicate all conflicts, that is, tokens belonging to two PREDICT(p) sets with the same LHS.

1. $P \longrightarrow S$ `$$`
2. $S \longrightarrow B$
3. $S \longrightarrow U$
4. $B \longrightarrow$ `if` C `then` B `else` B
5. $B \longrightarrow O$
6. $U \longrightarrow$ `if` C `then` S
7. $U \longrightarrow$ `if` C `then` B `else` U
8. $C \longrightarrow \mathtt{c}_i, i \geq 1$
9. $O \longrightarrow \mathtt{s}_i, i \geq 1$

Nonterminals: P, S, B, U, C, O; terminals: `if`, `then`, `else`, $\mathtt{c}_i$, $\mathtt{s}_i$, `$$`.

P -> S $$ { if c_i, s_i }.

S -> B { }

S -> U { s_i }.

4. (50pt) Write a Python program `comm_rm.py` to remove all comments from a C++ program. The program should work as follows:

```
comm_rm inputC.cpp inputC_rm.cpp
```

where `inputC.cpp` is any (correct) C++ program and `inputC_rm.cpp` is the same program with comments removed.