

## HW2

● Graded

Student

Adam Fenjiro

Total Points

28 / 52 pts

Question 1

Q1

2 / 8 pts

– 0 pts Correct

– 6 pts Point adjustment

Question 2

Q2

5 / 8 pts

– 0 pts Correct

– 3 pts Point adjustment

Question 3

Q3

11 / 20 pts

– 0 pts Correct

✓ – 1.5 pts Left Recursion - partially correct

✓ – 2.5 pts First/follow sets - partially correct

✓ – 2.5 pts Predict set - partially correct

✓ – 2.5 pts LL(1) parse table - partially correct

– 2 pts Is grammar LL(1)?

– Missing grammar rules after removing left recursion

Question 4

Q4

2 / 8 pts

– 0 pts Correct

– 6 pts Incomplete

– 6 pts Point adjustment

Question 5

Q5

8 / 8 pts

✓ - 0 pts Correct

- 2 pts Missing recursive check for non-terminal S
- 4 pts Missing recursive check for non-terminals S and T

Questions assigned to the following page: [1](#) and [2](#)

**Problem 1:**

FIRST()

- $\text{FIRST}(S') = \{s\}$
- $\text{FIRST}(S) = \{a, b, f\}$
- $\text{FIRST}(A) = \{b, \epsilon\}$
- $\text{FIRST}(B) = \{b, \epsilon\}$
- $\text{FIRST}(C) = \{b, f\}$

FOLLOW()

- $\text{FOLLOW}(S') = \{\$ \}$  (end of input marker)
- $\text{FOLLOW}(S) = \{\$ \}$  ( $S'$  is the start symbol)
- $\text{FOLLOW}(A) = \{c\}$  (followed by C in SC)
- $\text{FOLLOW}(B) = \{a, c, \$ \}$  (followed by A in ACB and end of input)
- $\text{FOLLOW}(C) = \{\$ \}$  (no more symbols after C)

PREDICT()

- $\text{PREDICT}(S) = \{a, b, f\}$
- $\text{PREDICT}(A) = \{\epsilon\}$
- $\text{PREDICT}(B) = \{\epsilon\}$
- $\text{Predict}(C) = \{b\}$

=> Yes, I think that the grammar is LL(1) because there are no conflicts in PREDICT() since each non-terminal and next input symbol has only one production to choose from.

**Problem 2:**

Here is the grammar when eliminating left recursion:

$S \rightarrow AcS'$

$S' \rightarrow abS' \mid \epsilon$

$A \rightarrow SabjKA' \mid jgAA' \mid hiA'$

$A' \rightarrow deA' \mid cjKA' \mid \epsilon$

Question assigned to the following page: [3](#)

**Problem 3:**

a.

 $\text{First}(B') = \{\epsilon, \text{or}\}$  $\text{First}(B) = \{\epsilon, \text{or}, \text{true}, \text{false}, ()\}$  $\text{First}(T') = \{\epsilon, \text{and}\}$  $\text{First}(T) = \{\text{true}, \text{false}, ()\}$  $\text{First}(C) = \{\text{true}, \text{false}, ()\}$  $\text{Follow}(B') = \{\$, ()\}$  $\text{Follow}(B) = \{\text{or}, \text{and}, \$, ()\}$  $\text{Follow}(T') = \{\$, ()\}$  $\text{Follow}(T) = \{\text{or}, \text{and}, \$, ()\}$  $\text{Follow}(C) = \{\text{or}, \text{and}, \$, ()\}$ 

b.

 $\text{PREDICT}(B) = \{\text{or}, \text{and}, \$, ()\}$  $\text{PREDICT}(B') = \{\text{or}, \text{and}, \$, ()\}$  $\text{PREDICT}(T) = \{\text{or}, \text{and}, \$, ()\}$  $\text{PREDICT}(T') = \{\text{or}, \text{and}, \$, ()\}$  $\text{PREDICT}(C) = \{\text{or}, \text{and}, \$, ()\}$ 

c.

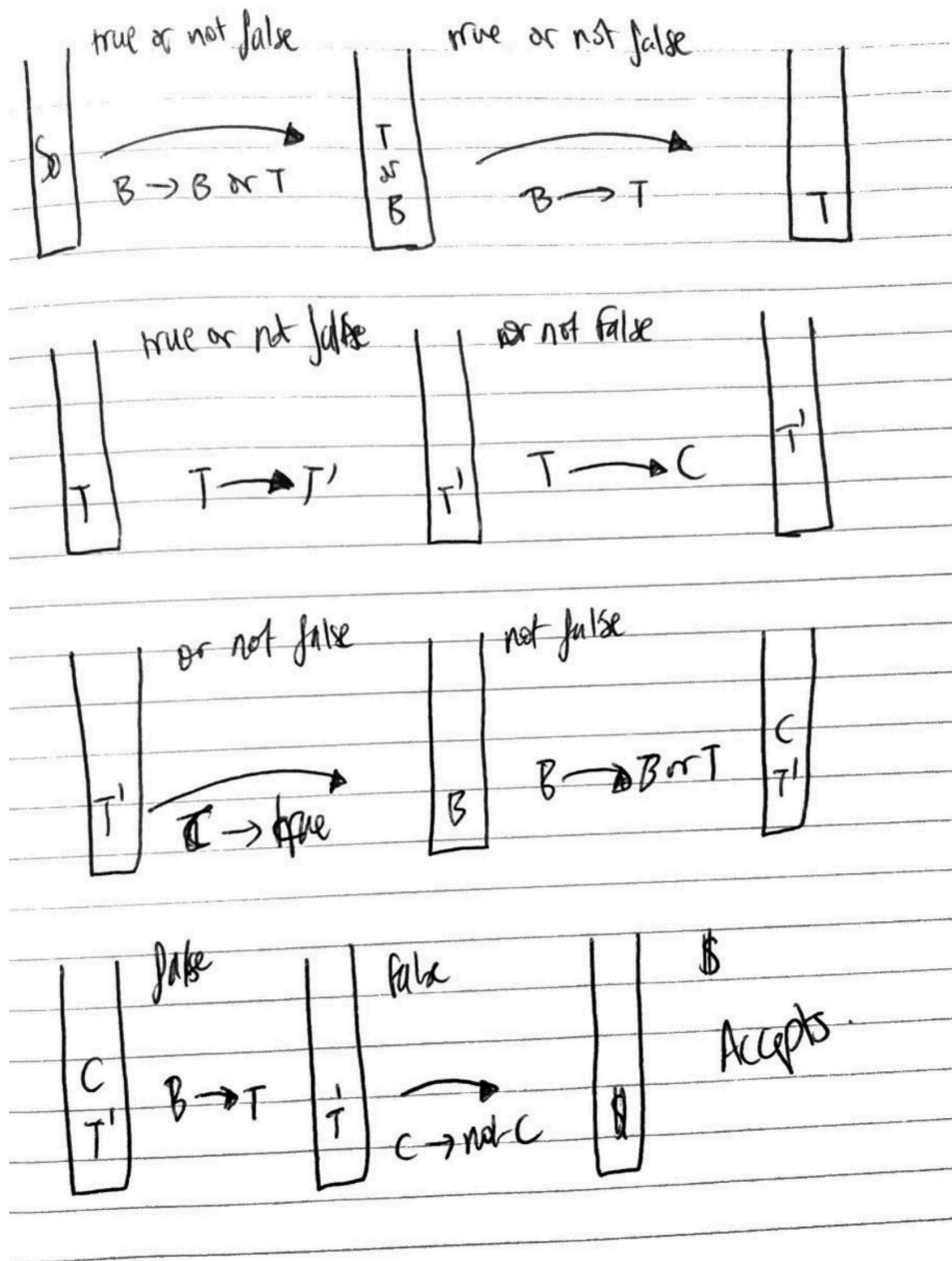
	A	B	C	D	E	F
S0		or	and	(	)	\$
S1	B	B'orT		(B)		\$
S2	B'	orTB'			orTB'	$\epsilon$
S3	T		TandC	(		\$
S4	T'		andCT'			$\epsilon$
S5	C	true	false	(B)		

d.

Since there is no conflicts in the parse table, we can say that the converted grammar is LL(1).

Question assigned to the following page: [4](#)

Problem 4:





Question assigned to the following page: [5](#)

### Problem 5:

```
C/C++
char current_char;

function main() {
    S();
}

function S() {
    if (current_char == '+') {
        match('+');
        S();
        T();
    }
    else if (current_char == '-') {
        match('-');
        S();
        T();
    }
    else if (current_char == 'a') match('a');
    else output error; //others chars need to be handled
}

function T() {
    if (current_char == 'V') {
        match('V');
        S();
    }
    else if (current_char == '^') {
        match('^');
        S();
    }
    else if (current_char == 'b') match('b');
    else output error; //others chars need to be handled
}
```