Caleb Klenda

Homework 3

3/23/2025

# Problem 1

**1.)**

* Nullable: A, E
* Terminal: a, b, c, d, e
* Non-terminal: S, A, B, E

**2.)**

**Atg File:**

COMPILER S

CHARACTERS

TOKENS

a = 'a' .

b = 'b' .

c = 'c' .

d = 'd' .

e = 'e' .

COMMENTS FROM "//" TO '\n'

IGNORE '\r'+'\n'+'\t'

PRODUCTIONS

S = A b | B c .

A = a A c | E | .

B = b B | d .

E = e E | .

END S.

**Trace.txt Sets**

S

first: a b d e

follow: EOF

A

first: a e

follow: b c

B

first: b d

follow: c

E

first: e

follow: b c

1.3)

Computing Select sets based on First and Follow sets:

* Select(S → A b) = {a, e, b}
* Select(S → B c) = {b, d}
* Select(A → a A c) = {a}
* Select(A → E) = {e, b}
* Select(A → EOF) = {b}
* Select(B → b B) = {b}
* Select(B → d) = {d}
* Select(E → e E) = {e}
* Select(E → EOF) = {b}

Both Select(S → A b) = {a, e, b} and Select(S → B c) = {b, d} contain b. This means the grammar is not LL(1).

**1.4)**

First, we can use the predictive parsing table to assume the first production (which is done since the grammar is not LL(1)).

Step 1: Initialize the Parser

* Stack: [S, EOF]
* Input: b d c EOF

Step 2: Process S

* Top of Stack: S
* Current Input Token: b
* Table Entry: S → A b (chosen because it is listed first in the productions)
* Stack After Expansion: [A, b, EOF]

Step 3: Process A

* Top of Stack: A
* Current Input Token: b
* Table Entry: A → E
* Stack After Expansion: [E, b, EOF]

Step 4: Process E

* Top of Stack: E
* Current Input Token: b
* Table Entry: E → EOF
* Stack After Expansion: [b, EOF]

Step 5: Match b

* Top of Stack: b
* Current Input Token: b
* Action: Match and consume b.
* Stack After Match: [EOF]
* Remaining Input: d c EOF

At this point, the stack is not empty, but more input remains, meaning the string is rejected

Running the input text through ATGSample also confirms this result:

A black background with white letters

AI-generated content may be incorrect.

While the grammar should accept the string, the generated parser does not resolve the LL(1) conflict correctly, and rejects the string.

# Problem 2