Spring'23 CPSC 323.02 Compilers & Languages
HW #3 [50 points]

Submission deadline: Sunday, May 7, 11:59 PM, submit it on Canvas

*Any HW content shall **NOT** be made publicly accessible without the written consent of the instructor.*

Top-down Parsers (*Reference: Textbook chapters 3.2, 3.3, 3.4*)

1. (10 points) Here we have a practical grammar for generating four-function expressions as below. Please remove all left recursions in the grammar (underline show your elimination steps).

   E $\rightarrow$ E + T | E – T | T
   T $\rightarrow$ T * F | T/F | F
   F $\rightarrow$ int | (E)

2. (20 points) Please determine the FIRST and FOLLOW from the revised grammar (*that does not contain left recursions*) in above Question 1.

3. (20 points) Construct the predictive-parser table for the revised grammar in above Question 1 using the FIRST and FOLLOW sets generated in above Question 2.

1. (10 points) Here we have a practical grammar for generating four-function expressions as below. Please remove all left recursions in the grammar (<u>show your elimination steps</u>).

$$E \rightarrow E + T \mid E - T \mid T$$
$$T \rightarrow T * F \mid T/F \mid F$$
$$F \rightarrow int \mid (E)$$

$$\underline{E \rightarrow E + T \mid E - T \mid T}$$

E → E +T (remove direct left-recursion)

E → TE`

E` → +TE` | −TE` | ε

$$\underline{T \rightarrow T*F \mid T/F \mid F}$$

T → T*F (remove direct left-recursion)

T → FT`

T` → *FT` | /FT` | ε

Final: E → TE`

E` → +TE` | −TE` | ε

T → FT`

T` → *FT` | /FT` | ε

F → int | (E)

2. (20 points) Please determine the FIRST and FOLLOW from the revised grammar (*that does not contain left recursions*) in above Question 1.

E → TE`

E` → +TE` | −TE` | ε

T → FT`

T` → *FT` | /FT` | ε

F → int | (E)

FIRST:

FIRST(E) = FIRST(T) = FIRST(F) = {int, (}

FIRST(E`) = {+, −, ε}

FIRST(T`) = {*, /, ε}

FOLLOW:

FOLLOW(E) = {$,)}

FOLLOW(E`) = {$,)}

FOLLOW(T) = {+, −, $,)} (after T in E → TE`)

FOLLOW(T`) = {+, −, $,)}

FOLLOW(F) = {*, /, +, −, $,)} (after F in T → FT`)

3. (20 points) Construct the predictive-parser table for the revised grammar in above Question 1 using the FIRST and FOLLOW sets generated in above Question 2.

$$Table[E`, \$] = Table[E,)] = \varepsilon$$
$$Table[T`, +] = Table[T`, \$] = Table[T`,)] = \varepsilon$$
$$Table[E, int] = Table[E, (] = TE`$$
$$Table[E`, +] = +TE`$$
$$Table[F, int] = int$$
$$Table[F, (] = (E)$$
$$Table[T`, /] = /FT`$$
$$Table[T`, *] = \varepsilon$$

	int	+	-	*	/	(	)	$
E	TE'					TE'		
E'		+TE'	-TE'				ε	ε
T	FT'					FT'		
T'		ε	ε	*FT'	/FT'		ε	ε
F	int					(E)		