Ex: Top-down parsing

- **Write a recursive descent recognizer for this grammar:**

A → int | '(' B ')'
B → A '+' B

```
type token = INT of int | LPAREN | RPAREN | PLUS | EOF

let rec parseA toklis = match (hd toklis) with




and parseB toklis =




and parse toklis = (parseA toklis = [EOF])
```

Top-down recognizer exercise

L $\rightarrow$ **E R**
E $\rightarrow$ **id**
R $\rightarrow$ **; L** $\mid \epsilon$

```
type token = IDENT of string | SEMIC | EOF
let rec parseL toklis =


and parseE toklis =



and parseR toklis =



and parse toklis = (parseL toklis = [EOF])
```

FIRST set examples

A $\rightarrow$ **id** | **'(' B ')'**
B $\rightarrow$ **int** | **A**

FIRST(A) = { }

FIRST(B) = { }

A $\rightarrow$ **int** | **'(' B ')'**
B $\rightarrow$ **A '+' B**

FIRST(A) = { }

FIRST(B) = { }

Ex: FIRST sets calculation

$E \rightarrow T \mid E + T$
$T \rightarrow P \mid T * P$
$P \rightarrow \textbf{id} \mid (E)$

$fsts_0$:

E	
T	
P	

$fsts_1$:

E	
T	
P	

$fsts_2$:

E	
T	
P	

$fsts_3$:

E	
T	
P	

$fsts_4$:

E	
T	
P	

Ex: FIRST sets calculation

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & \epsilon \mid \mathbf{+}\ E \\
T & \rightarrow & P\ T' \\
T' & \rightarrow & \epsilon \mid \mathbf{*}\ T \\
P & \rightarrow & \mathbf{id} \mid \mathbf{(}\ E\ \mathbf{)}
\end{array}
$$

$fsts_0$:

E	
E'	
T	
T'	
P	

$fsts_1$:

E	
E'	
T	
T'	
P	

$fsts_2$:

E	
E'	
T	
T'	
P	

$fsts_3$:

E	
E'	
T	
T'	
P	

$fsts_4$:

E	
E'	
T	
T'	
P	

Ex: FIRST sets calculation

$$
\begin{array}{lcl}
S & \rightarrow & A\ \mathbf{b} \\
A & \rightarrow & \mathbf{a} \mid B \mid \epsilon \\
B & \rightarrow & \mathbf{b} \mid \epsilon
\end{array}
$$

$fsts_0$:

S	
A	
B	

$fsts_1$:

s	
A	
B	

$fsts_2$:

S	
A	
B	

$fsts_3$:

S	
A	
B	

Ex: FOLLOW sets calculation

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & \epsilon \mid + \ E \\
T & \rightarrow & P\ T' \\
T' & \rightarrow & \epsilon \mid * \ T \\
P & \rightarrow & \textbf{id} \mid (\ E\ )
\end{array}
$$

$FIRST$:

E	**id, (**
E'	**•, +**
T	**id, (**
T'	**•, +**
P	**id, (**

$flws_0$:

E	
E'	
T	
T'	
P	

$flws_1$:

E	
E'	
T	
T'	
P	

$flws_2$:

E	
E'	
T	
T'	
P	

$flws_3$:

E	
E'	
T	
T'	
P	

Ex: FOLLOW sets calculation

$$
\begin{array}{lcl}
S & \rightarrow & A\ B\ A \mid \textbf{c}\ C \\
A & \rightarrow & \textbf{a} \mid \epsilon \\
B & \rightarrow & \textbf{b}\ D \mid \epsilon \\
C & \rightarrow & A\ D \mid \textbf{b} \\
D & \rightarrow & \textbf{a}\ A \mid \textbf{c}
\end{array}
$$

$FIRST$:

S	**a, b, c, •**
A	**a, •**
B	**b, •**
C	**a, b, c**
D	**a, c**

$flws_0$:

S	
A	
B	
C	
D	

$flws_1$:

S	
A	
B	
C	
D	

$flws_2$:

S	
A	
B	
C	
D	