6.1100 Spring 2024 Miniquiz #2

There are 2 pages. Please submit your answers on Gradescope by Feb 22nd, 2024, 11:59pm.

Name: 6.110 Staff

Email: 6.110-staff@mit.edu

1. Left recursion

Consider the following grammar. E, T, and F are non-terminals. **+**, **×**, **(**, **)**, and **id** are terminals.

$$
\begin{array}{lcl}
E & \rightarrow & T \mid E \mathbin{\mathbf{+}} T \\
T & \rightarrow & F \mid T \mathbin{\mathbf{\times}} F \\
F & \rightarrow & \underline{\mathbf{(}}\ E\ \underline{\mathbf{)}} \mid \mathbf{id}
\end{array}
$$

Eliminate left recursion in this grammar using the method taught in lecture.

Introduce new non-terminals as necessary.

We introduce new non-terminals: E' and T'. The updated grammar is as follows:

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & \mathbf{+}\ \mathrm{T}\ E' \mid \varepsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & \mathbf{\times}\ \mathrm{F}\ \mathrm{T'} \mid \varepsilon \\
F & \rightarrow & \underline{\mathbf{(}}\ E\ \underline{\mathbf{)}} \mid \mathbf{id}
\end{array}
$$

2. Left factoring

Consider the following grammar. P, E, and T are non-terminals. **+**, **[**, **]**, and **a** are terminals.

$$
\begin{array}{lcl}
P & \rightarrow & \underline{\mathbf{[}}\ E\ \underline{\mathbf{]}} \\
E & \rightarrow & T \mid T \mathbin{\mathbf{+}} E \\
T & \rightarrow & \mathbf{a}
\end{array}
$$

Perform left-factoring to eliminate common prefixes.

Here is the left-factored grammar:

$$
\begin{array}{lcl}
P & \rightarrow & \underline{\mathbf{[}}\ E\ \underline{\mathbf{]}} \\
E & \rightarrow & T\ E' \\
E' & \rightarrow & \mathbf{+}\ E \mid \varepsilon \\
T & \rightarrow & \mathbf{a}
\end{array}
$$

3. Constraint propagation

Consider the following grammar. *A*, *B*, *C*, *E*, and *P* are non-terminals. **[**, **]**, **(**, **)**, **a**, **b**, and **c** are terminals. *P* is the starting symbol.

$$
\begin{array}{lcl}
P & \rightarrow & E\ \mathbf{(}\ C\ \mathbf{)} \\
E & \rightarrow & A\ B \\
A & \rightarrow & \mathbf{a} \mid \varepsilon \\
B & \rightarrow & \mathbf{[}\ \mathbf{b}\ \mathbf{]} \mid \varepsilon \\
C & \rightarrow & \mathbf{c} \mid \varepsilon
\end{array}
$$

Compute the following.

a. The set **Nullable** = {*NT* : *NT* is a non-terminal that is able to derive ε}.

Nullable = {*A*, *B*, *C*, *E*}

b. For each non-terminal *NT*, the set
First(*NT*) = {*T* : *T* is a terminal that may appear at the left of a string derived from *NT*}

First(*P*) = {**(**, **[**, **a**}

First(*E*) = {**a**, **[**}

First(*A*) = {**a**}

First(*B*) = {**[**}

First(*C*) = {**c**}

Now, we define, for each non-terminal *NT*,

Follow(*NT*) = {*T* : *T* is a terminal that may appear directly after an expanded *NT* term in any valid derivation from the starting symbol}.

For example, we can show that **(** ∈ **Follow**(*A*). Consider the following derivation:

$P \rightarrow E\mathbf{(}C\mathbf{)} \rightarrow AB\mathbf{(}C\mathbf{)} \rightarrow \mathbf{a}B\mathbf{(}C\mathbf{)} \rightarrow \mathbf{a(}C\mathbf{)} \rightarrow \mathbf{a(c)}$.

The token **(** directly follows **a** which is expanded from *A*. (*B* is expanded into an empty string.)

c. *(Bonus problem; Optional)* Compute **Follow**(*NT*) for all non-terminal *NT*.

Follow(*P*) = {`eof`}

Follow(*E*) = {**(**}

Follow(*A*) = {**(**, **[**}

Follow(*B*) = {**(**}

Follow(*C*) = {**)**}