LL(1) Parsing, pt. 2

CS 440: Programming Languages and Translators, Spring 2020

A. Review: LL(1) Parsing for Grammars without ε rules

- So far we've looked at LL(1) grammars that don't have ε rules; they're generally easier to parse than languages with such rules.

Example 1:

- $S' \to S$ `$`, $S \to A\ B\ C\ S$, $A \to$ `a`, $B \to$ `b`, $C \to$ `c`, $S \to$ `s`.
  - The language of this grammar is (`abc`)*`s`, and the *First* sets are *First*(A) = {`a`}, *First*(B) = {`b`}, *First*(C) = {`c`}, and *First*(S) = *First*(S') = {`a`, `s`}. The *Predict* table shows that LL(1) parsing for this grammar is straightforward

NonT	**a**	**b**	**c**	**s**
S'	$S' \to S$ `$`			$S' \to S$ `$`
S	$S \to A\ B\ C\ S$			$S \to$ `s`
A	$A \to$ `a`			
B		$B \to$ `b`		
C			$C \to$ `c`	

Example 2:

- (To make life more interesting, let's) take the grammar from Example 1 and add the ε rules $B \to \varepsilon$ and $C \to \varepsilon$. Now LL(1) parsing is quite a bit more involved. The choices using ε for B and C are independent, so in effect, the rules for S are now $S \to A\ B\ C\ S \mid A\ C\ S \mid A\ B\ S \mid A\ S$. When trying to expand A, the next symbol can be `b`, `c`, or `s`.
- For the prediction table, *Predict*(X, `y`) should include rule $X \to \alpha$ if `y` $\in$ *First*(α) (and $\alpha \neq \varepsilon$ and `y` $\neq \varepsilon$). This is the same as prediction tables for grammars without ε rules. Predict(X, `y`) should include rule $X \to \varepsilon$ if `x` $\in$ *Follow*(X): informally, such `x` are what can appear after using X in a rule. We'll look at the details later, but
- *First*(S') = *First*(S) = {`a`}, *First*(A) = {`a`}, *First*(B) = {`b`, ε}, *First*(C) = {`c`, ε}. (The ε indicate that B and C $\to^* \varepsilon$.)
- *Follow*(S') = ∅, *Follow*(S) = {`$`}, *Follow*(A) = {`a`, `b`, `c`, `s`}, *Follow*(B) = {`a`, `c`, `s`}, *Follow*(C) = {`a`, `s`}.
- The *Predict* table:

NonT	**a**	**b**	**c**	**s**
S'	$S' \to S$ `$`			
S	$S \to A\ B\ C\ S$			$S \to$ `s`
A	$A \to$ `a`			
B	$B \to \varepsilon$	$B \to$ `b`	$B \to \varepsilon$	$B \to \varepsilon$
C	$C \to \varepsilon$		$C \to$ `c`	$C \to \varepsilon$

B. When do we use rule $X \to \alpha$ during LL(1) Parsing? (where $\alpha \neq \varepsilon$)

- As with grammars without ε rules, we use *First* sets to decide when to use a rule $X \to \alpha$.
  - I.e., rule $X \to \alpha \in$ *Predict*(X, `x`) if `x` $\in$ *First*(α). (And here, `x` $\neq \varepsilon$.)
- However, calculating the *First* sets is more complicated when ε rules are allowed.
- **Definition (original)**: *First*(α) = {`x` $\in T \mid \alpha \to^* $ `x` β for some $\beta \in (V \mid T)^*$)}. There are two cases depending on whether α begins with a terminal or nonterminal symbol.
- **Terminal symbol**: If $\alpha =$ `x` β, then *First*(α) = {`x`}. I.e., if something begins with a terminal symbol `x`, then that symbol makes up the *First* set.
- **Nonterminal symbol**: If $\alpha = Y\,\beta$, then for all rules $Y \to \gamma$, *First*($\alpha\,\beta$) $\supseteq$ *First*($\gamma\,\beta$). I.e., to find the *First* set for Y, we need the *First* set for all the rhs's of rules for Y.
- But there's a complicating factor, depending on whether $Y \to^* \varepsilon$ or not. If $Y \to^*$ `z`…, then for all β, $Y\,\beta \to^*$ `z`… β, so β is irrelevant here. However, if $Y \to^* \varepsilon$, then $Y\,\beta \to^* \varepsilon\,\beta$, so *First*($Y\,\beta$) $\supseteq$ *First*(β).
  - If $Y \not\to^* \varepsilon$ then for all β, *First*($Y\,\beta$) $\supseteq$ *First*(Y).
  - If $Y \to^* \varepsilon$ then for all β, *First*($Y\,\beta$) $\supseteq$ *First*(β).
  - (Note these two cases aren't mutually exclusive. E.g., when $Y \to$ `z` $\mid \varepsilon$)
- **Definition (extension)**: If $\alpha \to^* \varepsilon$, then $\varepsilon \in$ *First*(α).
  - This is done just as a way to store the answer to the question "Does $\alpha \to^* \varepsilon$?" One way is to have a separate table for it; here we'll add $\varepsilon \in$ *First*(α). (Note now *First*(X) $\subseteq T \cup \{\varepsilon\}$, not just T.)

C. When do we use a rule $X \to \varepsilon$ during LL(1) parsing?

- Recall *Predict(Y*, `x`) holds the rule to apply if the current nonterminal is Y and the current input symbol is `x`. In particular, rule $Y \to \alpha \in$ *Predict*(Y, `x`) if `x` $\in$ *First*($Y\,\beta$) where we can go from $S \to^* \ldots Y\,\beta$.
  - E.g., if $Y \to$ `x` Z, then `x` $\in$ *First*(Y). If $Y \to W$ and $W \to$ `x` Z, then `x` $\in$ *First*(W) and *First*(Y)
- But what if $Y \to \varepsilon$ is a rule? When should *Predict*(Y, `x`) contain $Y \to \varepsilon$?
  - For a concrete example, say we have $S \to Y$ `y` and $Y \to$ `x` $\mid \varepsilon$.
  - If our input is `x y`, we can use the derivation $S \to Y$ `y` $\to$ `x y` to parse it.
  - If our input is just `y`, then can use $S \to A$ `y` $\to \varepsilon$ `y` to parse it.
- More generally, if from the start symbol we can get to a form involving $Y\,\beta$, then if the next input symbol is in *First*(β), then using $Y \to \varepsilon$ makes sense.
  - So in addition to the earlier rule $X \to \alpha \in$ *Predict*(X, `x`) if `x` $\in$ *First*(X), we have
  - (New) $X \to \varepsilon \in$ *Predict*(X, `x`) if $X \to^* \varepsilon$ and `x` $\in$ *First*(β) for some β where $S \to^* \ldots X\,\beta$.
  - (Side comment: $X \to^* \varepsilon$ doesn't require $X \to \varepsilon$, e.g., when $X \to Z$ and $Z \to \varepsilon$. The rule $Z \to \varepsilon$ will go into *Predict*(Z, …) when we worry about Z.)

D. Using Follow sets to simplify the decision involving putting $X \to \varepsilon$ in the prediction table

- Currently we have $X \to \varepsilon \in$ *Predict*(X, `x`) if $X \to^* \varepsilon$ and `x` $\in$ *First*(β) for some β where $S \to^* \ldots X\,\beta$.

- The problem here is it requires that we know all possible β than can follow X in a derivation from S and there can be an infinite number of them.
- However, we can use a simpler algorithm for seeing if $X \to \varepsilon$ is appropriate: Instead of calculating *First*(β) for all β, we can ask what terminal symbols can follow a use of X. There are only a finite number of nonterminals, so there are only a finite number of *Follows*(…) calculations to make.
- For a small example, say we have $S \to A\ B$, $A \to \varepsilon$, $B \to \mathtt{b}\ B \mid \mathtt{c}\ B \mid \varepsilon$.
  - If we ask "What are all the β where $S \to^* A\ \beta$?" the answer is $(\mathtt{b} \mid \mathtt{c})^* - \varepsilon$, which is of infinite size.
    - (We don't include ε in any *Follow* set so $A\ \beta \to^* A\ \varepsilon$ is not a worry, though $\beta \to^* \varepsilon$ does have to be taken into account elsewhere.)
  - On the other hand, if we ask "What terminal symbols can follow A in a derivation?", the answer $\{\mathtt{b}, \mathtt{c}\}$ can be calculated more easily.
- **Definition**: $\mathit{Follow}(X) = \{\mathtt{y} \in T \mid \text{for some } \beta',\ S \to^* \ldots X\ \mathtt{y}\ \beta'\}$. I.e., the follow set of A is the set of terminal symbols that can follow an appearance of A in some derivation. (This might not be a leftmost derivation, but that's okay as long as the grammar isn't ambiguous.)
- The advantage of looking at follow sets vs first sets is that we only need *Follow*(…) for each nonterminal whereas for *First*(…), we need an entry for each $A\ \beta$ (which are possibly infinite in number).
- So for adding $X \to \varepsilon$ to the *Predict* table, instead of
  - If from S we can derive a form $X\ \beta$ and $\mathtt{x} \in \mathit{First}(\beta)$), then rule $X \to \varepsilon \in \mathit{Predict}(X, \mathtt{x})$

  we'll use
  - Rule $X \to \varepsilon \in \mathit{Predict}(X, a)$ for all $a \in \mathit{Follow}(A)$.

E. Calculating Follow sets

- So how do we calculate $\mathit{Follow}(X) = \{\mathtt{y} \in T \mid \text{for some } \beta',\ S \to^* w\ X\ \mathtt{y}\ \beta'\}$?
- For any rule $Z \to \ldots X \ldots$ that has an X on its rhs, there are roughly three cases to *Follow*(X)
  - Case 1: If $Z \to \ldots X\ \mathtt{y}\ \beta'$ then $\mathtt{y} \in \mathit{Follow}(X)$
  - Case 2: If $Z \to \ldots X\ C\ \beta'$ then $\mathit{First}(C) - \varepsilon \subseteq \mathit{Follow}(X)$
    - But if $C \to^* \varepsilon$, then (in effect) we have to consider $Z \to \ldots X\ \beta'$, which leads to the two cases so far unless β' begins with a nonterminal that $\to^* \varepsilon$, etc. It's possible we need to iterate case 2, which leads to case 3:
  - Case 3: If $Z \to \ldots X\ C_1\ C_2 \ldots C_n$ where every $C_i \to^* \varepsilon$, then $\mathit{Follow}(Z) \subseteq \mathit{Follow}(X)$.
    - (Note this can include the case where $n = 0$, so we can generalize this to if $Z \to \ldots X\ \gamma$ where $\gamma \to^* \varepsilon$, then $\mathit{Follow}(Z) \subseteq \mathit{Follow}(X)$.)
  - To understand this case, here's a derivation showing how anything that follows Z can also follow X:

    $S' \to^* \ldots Z\ \delta \to \ldots \ldots X\ \beta\ \delta \to^* \ldots X\ \delta$

Parsing prediction table when ε-rules exist

- There are now two kinds of entries in the parsing table
  - $\mathit{Predict}(X, \mathtt{x})$ includes rule $X \to \alpha$ (where $\alpha \neq \varepsilon$) if $\mathtt{x} \in \mathit{First}(\alpha)$.

- $Predict(X, \texttt{x})$ includes rule $X \to \varepsilon$ if $\texttt{x} \in Follow(X)$.
- Again, if any table entry contains >1 rule, the grammar is not $LL(1)$.
- And if a table entry is empty, then that nonterminal / terminal character causes a parse error

Example 3:

- (Revisiting Example 2) $S' \to S\ \$$, $S \to A\ B\ C\ S$, $A \to \texttt{a}$, $B \to \texttt{b} \mid \varepsilon$, $C \to \texttt{c} \mid \varepsilon$, $S \to \texttt{s}$.
- For the *First* sets, we can calculate $First(S') = First(S) = \{\texttt{a}, \texttt{s}\}$, $First(A) = \{\texttt{a}\}$, $First(B) = \{\texttt{b}, \varepsilon\}$, and $First(C) = \{\texttt{c}, \varepsilon\}$.
  - For $First(S)$, $S \to \texttt{s}$ implies $\texttt{s} \in First(S)$, and $S \to A\ B\ C\ S$ implies $First(A) \subseteq First(S)$.
    - For $First(A)$, $A \to \texttt{a}$, implies $\texttt{a} \in First(S)$ Since $A \nrightarrow^* \varepsilon$, we don't have to consider $First(B\ C\ S)$.
    - There's no other $A \to \ldots$ rule, so $First(A) = \{\texttt{a}\}$ which implies $First(S) = \{\texttt{a}, \texttt{s}\}$.
  - For $First(B)$, $B \to \texttt{b} \mid \varepsilon$ implies $\{\texttt{b}, \varepsilon\} \subseteq First(B)$. There are no other $B \to \ldots$ rules, so $\{\texttt{b}, \varepsilon\} = First(B)$.
  - For $First(C)$, $C \to \texttt{c} \mid \varepsilon$ implies $\{\texttt{c}, \varepsilon\} \subseteq First(C)$. There are no other $C \to \ldots$ rules, so $\{\texttt{c}, \varepsilon\} = First(C)$.
  - For $First(S')$, $S' \to S\ \$$ implies $First(S) \subseteq First(S')$. There are no other S' rules, so $First(S') = \{\texttt{a}, \texttt{s}\}$.
- For the *Follow* sets,
  - $Follow(S') = \varnothing$ because S' doesn't appear on the rhs of any rule.
  - $Follow(S) = \{\$\}$ because $S' \to S\ \$$ is the only rhs that S appears in.
  - For $Follow(C)$, since C appears in $S \to A\ B\ C\ S$, we have $Follow(C) \subseteq First(S) - \varepsilon \subseteq \{\texttt{a}, \texttt{s}\}$.
    - Since $S \nrightarrow^* \varepsilon$, we don't have to worry about $Follow(C)$ including $Follow(S)$ (the lhs of the rule). Since C only appears in this one rule, we can change the $\subseteq$ to $=$ in $Follow(C) \subseteq \{\texttt{a}, \texttt{s}\}$.
  - For $Follow(B)$, since B appears in $S \to A\ B\ C\ S$, we have $Follow(B) \subseteq First(C\ S) - \varepsilon \subseteq (First(C) - \varepsilon) \cup First(S) - \varepsilon = \{\texttt{a}, \texttt{c}, \texttt{s}\}$. (Since $C \to^* \varepsilon$, $Follow(B)$ has to include $First(S)$.)
    - Since $C\ S \nrightarrow^* \varepsilon$, we don't have to worry about $Follow(B)$ including $Follow(C\ S)$ (the lhs of the rule). Since B only appears in the one rule, we can change the $\subseteq$ to $=$ in $Follow(B) \subseteq \{\texttt{a}, \texttt{c}, \texttt{s}\}$
  - For $Follow(A)$, since A appears in $S \to A\ B\ C\ S$, we have $Follow(A) \subseteq First(B\ C\ S) - \varepsilon$. Since $B \to^* \varepsilon$ and $C \to^* \varepsilon$, $First(B\ C\ S) \subseteq (First(B) - \varepsilon) \cup (First(C) - \varepsilon) \cup (First(S) - \varepsilon) = \{\texttt{b}\} \cup \{\texttt{c}\} \cup \{\texttt{a}, \texttt{s}\} = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{s}\}$.
- So $Follow(S') = \varnothing$, $Follow(S) = \{\$\}$, $Follow(A) = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{s}\}$, $Follow(B) = \{\texttt{c}, \texttt{s}\}$, and $Follow(C) = \{\texttt{s}\}$.
- The *Predict* table is below. Since only B and C have $\to \varepsilon$ rules, it's only their follow sets that we need.

NonT	**a**	**b**	**c**	**s**
S'	$S' \to S\ \$$			
S	$S \to A\ B\ C\ S$			$S \to \texttt{s}$
A	$A \to \texttt{a}$			
B	$B \to \varepsilon$	$B \to \texttt{b}$	$B \to \varepsilon$	$B \to \varepsilon$
C	$C \to \varepsilon$		$C \to \texttt{c}$	$C \to \varepsilon$

Example 4: Sample Calculation of *First* and *Follow* (from the textbook)

- *Grammar*: $S \to$ `a` $A\ B$ `b`, $A \to$ `c` | ε, $B \to$ `d` | ε
- *First sets*
  - *First*(S) ⊇ *First*(`a` $A\ B$ `b`) = {`a`}
  - *First*(A) ⊇ *First*(`c`) ∪ *First*(ε) = {`c`, ε}
  - *First*(B) ⊇ *First*(`d`) ∪ *First*(ε) = {`d`, ε}
  - These are the only calculations to do, so the ⊇ above can be turned into equality.
- *Follow sets*
  - *Follow*(B): From $S \to$ `a` $A\ B$ `b`, the B `b` part tells us
    - *Follow*(B) ⊇ {`b`}
  - *Follow*(A): From $S \to$ `a` $A\ B$ `b`, the $A\ B$ `b` tells us
    - *Follow*(A) ⊇ *First*(B `b`) - {ε} = {`d`, ε} - {ε} = {`d`}
    - Since $B \to^* ε$, we also have *Follow*(A) ⊇ *First*(`b`) = {`b`}
    - Altogether, *Follow*(A) ⊇ {`b`, `d`}
  - These are the only calculations to do, so the ⊇ above can be turned into equality
- **Prediction Table** (empty entries indicate errors)

NonT	`a`	`b`	`c`	`d`
S	$S \to$ `a` $A\ B$ `b`			
A		$A \to ε$	$A \to$ `c`	$A \to ε$
B		$B \to ε$		$B \to$ `d`

General observation

- If *First*(X) and *Follow*(X) have a symbol `x` in common, that's not a problem if there's no $X \to ε$ rule.
- But if there is an $X \to ε$ rule, then the grammar is ambiguous because we'd want Predict(X, x) to include both $X \to$ non-ε and $X \to ε$ rules.

Activity Questions for Lecture 14

1. Give the *First* and *Follow* sets and *Predict* table for the grammar below. Is the grammar LL(1)?

   $S \to A$ `$`

   $A \to$ `a` B

   $B \to C\ D$ `e`

   $C \to$ `c` | D

   $D \to$ `d` | ε

2. Give the *First* and *Follow* sets and *Predict* for the grammar below. Is the grammar LL(1)?

   $P \to Q\ R\ S$ `$`

   $Q \to$ `q` | ε

   $R \to$ `r` | ε

   $S \to$ `s` | ε

Solutions to Activity Questions for Lecture 14

1. (Calculate *First* & *Follow* and LL prediction table.)

Here's the reasoning behind the *First* and *Follow* set contents with each rule and its implications.

$S \to A$ `$`	$First(S) \supseteq First(A)$; `$` $\in Follow(A)$
$A \to$ `a` B	`a` $\in First(A)$; $Follow(A) \subseteq Follow(B)$
$B \to C\ D$ `e`	$First(B) \supseteq First(C)$ - ε; $First(B) \supseteq Follow(C)$ [because $C \to^* \varepsilon$); $Follow(C) \supseteq First(D)$ - ε; `e` $\in Follow(C)$ (because $D \to^* \varepsilon$); `e` $\in Follow(D)$
$C \to$ `c` | D	`c` $\in First(C)$; $First(C) \supseteq First(D)$, $Follow(C) \subseteq Follow(D)$
$D \to$ `d` | ε	`d` $\in First(C)$; $\varepsilon \in First(D)$

So we get

$First(S) = \{$`a`$\}$, $Follow(S) = \varnothing$

$First(A) = \{$`a`$\}$, $Follow(A) = \{$`$`$\}$

$First(B) = \{$`c`, `d`, `e`$\}$, $Follow(B) = \{$`$`$\}$

$First(C) = \{$`c`, `d`, $\varepsilon\}$, $Follow(C) = \{$`d`, `e`$\}$

$First(D) = \{$`d`, $\varepsilon\}$, $Follow(D) = \{$`d`, `e`$\}$

(`d` $\in Follow(D)$ because of $C \to D$ with … $\to C\ D$ … and $D \to$ `d`)

The prediction table is

Nonterminal	`a`	`c`	`d`	`e`
S	$S \to A$ `$`			
A	$A \to$ `a` B			
B		$B \to C\ D$ `e`	$B \to C\ D$ `e`	$B \to C\ D$ `e`
C		$C \to$ `c`	$C \to D$	$C \to D$
D			$D \to$ `d` $D \to \varepsilon$	$D \to \varepsilon$

The grammar is not LL(1) because of the conflict between $D \to$ `d` and $D \to \varepsilon$ on input `d`.

2. (Grammar $P \to Q\ R\ S\ \$$, $Q \to$ q | ε, $R \to$ r | ε, $S \to$ s | ε)

The First and follow sets and predict table are below. The grammar is LL(1).

Nonterminal	First sets	Follow sets
P	q r s $	∅
Q	q ε	r s $
R	r ε	s $
S	s ε	$

Nonterminal	q	r	s	$
P	$P \to Q\ R\ S\ \$$	$P \to Q\ R\ S\ \$$	$P \to Q\ R\ S\ \$$	
Q	$Q \to$ q	$Q \to \varepsilon$	$Q \to \varepsilon$	$Q \to \varepsilon$
R		$R \to$ r	$R \to \varepsilon$	$R \to \varepsilon$
S			$S \to$ s	$S \to \varepsilon$

