Tutorial 6

- Top-down Parsing (LL(1))
- ε-NFAs to DFAs

LL(1) Parsing

- Inputs: CFG $G = (N, \Sigma, P, S)$ & input string $x \in \Sigma^*$
- finds derivation from start symbol S to input string x (ie: $S \Rightarrow^* x$)
  - ↳ let α_i be derivation i from $S \Rightarrow \dots \Rightarrow \alpha_i \Rightarrow \dots \Rightarrow x$,

    rule we apply

    $i \geq 0$, $\alpha_i \in (N \cup \Sigma)^*$ (origin: $A \in N$, $A \rightarrow \boxed{\alpha_i} \in P$)
  - ↳ goal: find $S \Rightarrow \alpha_0 \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow x$ to prove

    $x \in L(G)$ or prove $x \notin L(G)$

    $L(G)$ = "Language of the grammar G"
- LL(1) ⟶ one symbol lookahead
  - ↳ parser makes decision about what rule α_i to apply based on <u>only the next symbol in the input</u>
  - ↳ ☆ LL(1) parsing doesn't always work due to this limitation!

Predict Table

- Used to determine if LL(1) parsing will work for G
- let $A \in N$

  $a, b, c \in \Sigma$

  $\alpha, \beta \in (N \cup \Sigma)^*$ ⟵ strings of non-terminals & terminals

  $A \rightarrow \alpha \in P$

- Predict(A, a) creates a set of production rules
  - ↳ Contains rule $A \to \alpha$ if:
    1. α can derive a string whose 1st symbol is a
       - ↳ We want to see a string $A \to$ "$a\beta$"
    2. α " " ε and it is possible for a to immediately follow A in the derivation
       - ↳ "get rid" of A, $A \Rightarrow^*$ "$\varepsilon \alpha \beta$"
- Formally:

  $\text{First}(\alpha) = \{ b \mid \alpha \Rightarrow^* b\beta \text{ for some } \beta \}$

  $\text{Follow}(A) = \{ c \mid S' \Rightarrow^* \alpha A c \beta \text{ for some } \alpha, \beta \}$

  $\text{Nullable}(\alpha) =$ true if $\alpha \Rightarrow^* \varepsilon$, false otherwise

  $\text{Predict}(A, \alpha) = \{ A \to \alpha \mid (a \in \text{First}(\alpha)) \text{ or } (\text{Nullable}(\alpha) \text{ and } a \in \text{Follow}(A)) \}$

- ☆ if Predict(A, α) contains ≤ 1 rule for all pairs (A, a), we say G is LL(1) & we can use LL(1) to parse G

eg: Consider the CFG:

① $S' \to \vdash S \dashv$

② $S \to a X Y b$

③ $S \to X Y$

④ $X \to p X$

⑤ $X \to \varepsilon$

⑥ $Y \to q$

⑦ $Y \to \varepsilon$

Compute the predict table for this grammar.

- Tip: Compute in the order Nullable, First then Follow
  - ↳ Use algos from class or intuition!

Nullable(A)

- $\forall A \in N$, track what rules can be used to "nullify" A (ie: $A \Rightarrow^* \varepsilon$)
  - ↳ Nullable(X) = Nullable(Y) = true from rules ⑤ & ⑦
  - ↳ " $A = S$, Nullable(A) = true since $S \Rightarrow XY$ by ③ & X, Y are nullable
  - ↳ Nullable(S') = false since S' only has rule ①, $S' \Rightarrow \vdash S \dashv$, with terminals $\vdash, \dashv \in \Sigma$

$A \in N$	Nullable(A)	Nullable Rules
S'	False	none
S	True	③
X	True	⑤
Y	True	⑦

First(A)

- $\forall A \in N$, look at the 1^{st} symbol on the right hand side of each rule that expands A
  - ↳ $S' \to \vdash S \dashv$ by ①, $\therefore$ First$(S') = \{\vdash\}$
    - → add ① to Predict$(S', \vdash)$
  - ↳ X is expanded by ④ & ⑤
    - → ④ : $X \to pX$, $\therefore$ $p \in$ First(X), add ④ to Predict(X, p)
    - → ⑤ : $X \to \varepsilon$, ignore
    - → $\therefore$ First$(X) = \{p\}$
  - ↳ Y is expanded by ⑥ & ⑦
    - → ⑥ : $Y \to q$, $\therefore$ $q \in$ First(Y), add ⑥ to Predict(Y, q)
    - → ⑦ : $Y \to \varepsilon$, ignore
    - → $\therefore$ First$(Y) = \{q\}$
  - ↳ S is expanded by ② & ③
    - → ② : $S \to aXYb$, $\therefore$ $a \in$ First(S), add ② to Predict(S, a)
    - → ③ : $S \to XY$ ($S \Rightarrow XY \Rightarrow pXY$ & $S \Rightarrow XY \Rightarrow Y \Rightarrow q$)
      - ↳ Anything in First(X) is in First(S) (ie: First$(X) \subseteq$ First(S))
      - ↳ $\because$ Nullable(X) = true, Anything in First(Y) is in First(S) (ie: First$(Y) \subseteq$ First(S))
      - ↳ thus $p, q \in$ First(S), add ③ to Predict(S, q) & Predict(S, p)
    - → $\therefore$ First$(S) = \{a, p, q\}$

$A \in N$	First(A)
S'	$\{\vdash\}$
S	$\{a, p, q\}$
X	$\{p\}$
Y	$\{q\}$

Partially-Filled Predict Table:

Predict	$\vdash$	$\dashv$	a	b	p	q
S'	1					
S			2		3	3
X					4	
Y						6

Follow (A)

- $\forall A \in N$, look at rules where A appears on the right hand side & figure out what symbols could appear after A
- If we add $a \in \Sigma$ to Follow(A), add all <u>nullable</u> rules for A to Predict(A, a) $\leftarrow$ lets us get rid of A to get a on the LHS
  - ↳ Follow$(S') = \emptyset$ since S' doesn't appear in the RHS of any rule
  - ↳ Follow $(S) = \{\dashv\}$ since by 1 we get $S' \to \vdash S \dashv$ & Nullable (S) = true.
    by 3
    $\longrightarrow$ add 3 to Predict$(S, \dashv)$

↳ Follow $(Y) = \{b, \dashv\}$

→ Y on RHS of $S \to aXYb$ & is followed by b, $b \in$ Follow(Y)

→ Y " " " $S \to XY\varepsilon$, anything in Follow(S) is in Follow(Y).

thus Follow$(S) = \{\dashv\} \subseteq$ Follow(Y)

→ ∴ Follow$(Y) = \{b\} \cup$ Follow$(S) = \{b, \dashv\}$, since

Nullable(Y) = true add ⑦ to Predict(Y, b) & Predict$(Y, \dashv)$

↳ Follow $(X) = \{b, q, \dashv\}$

→ $S \to aXYb$, anything in First$(Y) \subseteq$ Follow(X)

→ $S \to aXYb$, $b \in$ Follow(X) ∵ Nullable(Y) = true

→ $S \to XY\varepsilon$, ∵ Nullable(Y) = true, Follow$(S) \subseteq$ Follow(X)

→ $X \to pX\varepsilon$, no info on Follow(X)

→ ∴ Follow(X) = First$(Y) \cup \{b\} \cup$ Follow$(S) = \{b, q, \dashv\}$

↳ add nullable rule ⑤ to Predict(X, b), Predict(X, q) & Predict$(X, \dashv)$

$A \in N$	Follow
S'	$\varnothing$
S	$\{\dashv\}$
X	$\{b, q, \dashv\}$
Y	$\{b, \dashv\}$

Final Predict Table:

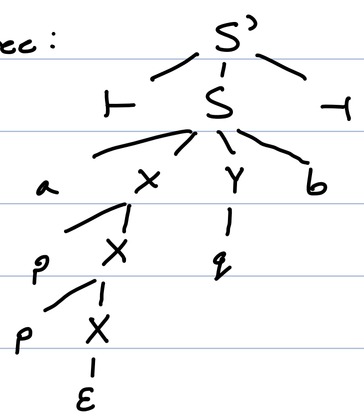
Predict	$\vdash$	$\dashv$	a	b	p	q
S'	1					
S		3	2		3	3
X		5		5	4	5
Y		7		7		

eg: Parse $\vdash appqb \dashv$ using the created predict table

Stack	Matched	Input	Action
S'	ε	$\vdash appqb\dashv$	Predict$(S', \vdash)$ = Apply ① pop push $S' \to \vdash S \dashv$
$\vdash S \dashv$	ε	$\cancel{\vdash} appqb\dashv$	Match $\vdash$ (Pop $\vdash$, consume $\vdash$ from input)
$S \dashv$	$\vdash$	$appqb\dashv$	Predict(S, a) = Apply ②, $S \to aXYb$
$aXYb\dashv$	$\vdash$	$\cancel{a}ppqb\dashv$	Match a
$XYb\dashv$	$\vdash a$	$ppqb\dashv$	Predict(X, p) = Apply ④, $X \to pX$
$pXYb\dashv$	$\vdash a$	$\cancel{p}pqb\dashv$	Match p
$XYb\dashv$	$\vdash ap$	$pqb\dashv$	Predict(X, p) = Apply ④, $X \to pX$
$pXYb\dashv$	$\vdash ap$	$\cancel{p}qb\dashv$	Match p
$XYb\dashv$	$\vdash app$	$qb\dashv$	Predict(X, q) = Apply ⑤, $X \to \varepsilon$
$Yb\dashv$	$\vdash app$	$qb\dashv$	Predict(Y, q) = Apply ⑥, $Y \to q$
$qb\dashv$	$\vdash app$	$\cancel{q}b\dashv$	Match q
$b\dashv$	$\vdash appq$	$\cancel{b}\dashv$	Match b
$\dashv$	$\vdash appqb$	$\dashv$	Match $\dashv$
ε	$\boxed{\vdash appqb\dashv}$	ε	Accept! (Stack = Input = ε)
$S' \Rightarrow^* \vdash appqb\dashv$	As desired!		

- The Stack rules from top (S') to bottom (ε) give us the leftmost derivation of $S' \Rightarrow^* \vdash appqd \dashv$

- Parse Tree:

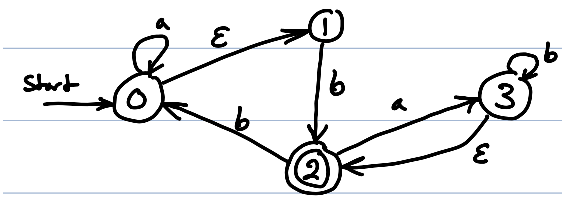


ε-NFAs to DFAs

- Similar to converting an NFA to a DFA, but you need to account for ε-transitions when figuring out what states are reachable from a given set of states
  - ↳ Initial State: $\{q_0\} \cup \{q \mid q$ reached by taking ≥ 1 ε-trans from $q_0\}$
  - ↳ for state q_i, find all transitions $\forall$ symbols a $(\delta(q_i, a))$ & figure out what other states, q_j, are reachable by ≥ 1 ε-transitions

Finding the ε-closure $\forall$ states q_i, $i \geq 0$

eg: Convert the ε-NFA to a DFA



Transition table $(\delta(q,a)\ \forall q \in S$ and $a \in \Sigma)$ for the NFA

		a	b	ε ← Seperate column for clarity
α	0	$\{0\}$	$\emptyset$	$\{1\}$ ← $0 \xrightarrow{\varepsilon} 1$
	1	$\emptyset$	$\{2\}$	$\emptyset$
	2	$\{3\}$	$\{0\}$	$\emptyset$
β	3	$\emptyset$	$\{3\}$	$\{2\}$ ← $3 \xrightarrow{\varepsilon} 2$

- Now we want to create the NFA to DFA translation (remove ε)
  - ↳ ☆ Before adding a set to the table, add in <u>all states</u> that are reachable <u>by states in the set</u> by ≥ 1 ε-transitions.

	a	b
$\{0,1\}$	$\{0,1\}$	$\{2\}$
$\{2\}$	$\{2,3\}$	$\{0,1\}$
$\{2,3\}$	$\{2,3\}$	$\{0,1,2,3\}$
$\{0,1,2,3\}$	$\{0,1,2,3\}$	$\{0,1,2,3\}$

← ∵ we start in 0 or 1 by ε (∝ ref)

← Add $\{2,3\}$, from 2 given a end in 2 or 3

← on b, $\{2,3\}$ goes to $\{0,3\}$, account for ε-trans in 0 & 3 gives us $\{0,1,2,3\}$