Nondeterministic Bottom–Up Parsing

Recap: Leftmost analysis with NTA

- We have seen the nondeterministic Top-Down parsing automaton that gives the leftmost analysis of an input
- Example

$$E \to E + T \mid T \qquad (1, 2)$$
$$T \to T * F \mid F \qquad (3, 4)$$
$$F \to (E) \mid a \mid b \qquad (5, 6, 7)$$

on input $(a) * b$

• Initial state	$((a) * b,\ E,\ \varepsilon)$
• Rule 2 to expand E	$((a) * b,\ T,\ 2)$
• Rule 3 to expand T	$((a) * b,\ T * F,\ 23)$
• Rule 4 to expand T	$((a) * b,\ F * F,\ 234)$
• Rule 5 to expand E	$((a) * b,\ (E) * F,\ 2345)$
• Match the "("	$(a) * b,\ E) * F,\ 2345)$
• Rule 2 to expand E	$(a) * b,\ T) * F,\ 23452)$
• Rule 4 to expand T	$(a) * b,\ F) * F,\ 234524)$
• Rule 6 to expand F	$(a) * b,\ a) * F,\ 2345246)$
• Match "a"	$() * b,\ ) * F,\ 2345246)$
• Match ")"	$(* b,\ * F,\ 2345246)$
• Match "*"	$(b,\ F,\ 2345246)$
• Rule 7 to expand F	$(b,\ b,\ 23452467)$
• Match "b"	$(\varepsilon,\ \varepsilon,\ 23452467)$

Bottom-up parsing

- Idea: Parse from the leafs of the syntax tree toward the root
- Example

$$E \to E + T \mid T \qquad (1, 2)$$
$$T \to T * F \mid F \qquad (3, 4)$$
$$F \to (E) \mid a \mid b \qquad (5, 6, 7)$$

  on input $(a) * b$
- (video)

Nondeterministic Bottom-Up Automaton (NBA)

- Similar to NTA, we define the NBA for $G =< \Sigma, N, P, S >$ by
  - Input alphabet Σ
  - Pushdown alphabet $X = N \cup \Sigma$
  - Output alphabet $U =$ the rule numbers 1,2,3,...
  - States $\Sigma^* \times X^* \times U^*$
  - Two types of transitions for $w \in \Sigma^*, \alpha \in X, z \in U$:
    - **Reducing** using rule $A \to \beta$ with number i:
      $$(w, \alpha\beta, z) \to (w, \alpha A, z\ i)$$
    - **Shifting** of terminal symbol $a \in \Sigma$:
      $$(aw, \alpha, z) \to (w, \alpha a, z)$$
  - Initial state $(w, \varepsilon, \varepsilon)$ for $w \in \Sigma^*$
  - Final state (ε, S, u) where $u \in U^*$
- **Theorem (without proof)**: Running the NBA on input w gives the reversed rightmost analysis u of w

Example: Reversed Rightmost Analysis with NBA

$$E \to E + T \mid T \qquad (1, 2)$$
$$T \to T * F \mid F \qquad (3, 4)$$
$$F \to (E) \mid a \mid b \qquad (5, 6, 7)$$

- Running the NBA on $w = (a) * b$

Initial state	$((a) * b,\ \varepsilon,\ \varepsilon)$
Shift	$(a) * b,\ (,\ \varepsilon)$
Shift	$() * b,\ (a,\ \varepsilon)$
Rule 6 to reduce a to F	$() * b,\ (F,\ 6)$
Rule 4 to reduce F to T	$() * b,\ (T,\ 64)$
Rule 2 to reduce T to E	$() * b,\ (E,\ 642)$
Shift	$(* b,\ (E),\ 642)$
Rule 5 to reduce (E) to F	$(* b,\ F,\ 6425)$
Rule 4 to reduce F to T	$(* b,\ T,\ 64254)$
Shift	$(b,\ T *,\ 64254)$
Shift	$(\varepsilon,\ T * b,\ 64254)$
Rule 7 to reduce b to F	$(\varepsilon,\ T * F,\ 642547)$
Rule 3 to reduce $T * F$ to T	$(\varepsilon,\ T,\ 6425473)$
Rule 2 to reduce T to E	$(\varepsilon,\ E,\ 64254732)$

Nondeterminism

- Like NTAs, NBA parsing can be nondeterministic:
  - If we have a rule $A \to a$ and the state $(bw, \alpha a, z)$, should we reduce a to A or shift b?
  - If we have rules $A \to ab$ and $B \to b$ and the state $(w, \alpha ab, z)$, should we reduce ab to A or b to B?
  - If we have rules $A \to a$ and $B \to a$ and the state $(w, \alpha a, z)$, should we reduce a to A or to B?
  - If we have rule $A \to S$ (where S is the initial symbol) and the state (ε, S, z), should we stop parsing or reduce S to A?
- The last case can be easily fixed:

  For grammars like

$$S \to \cdots$$
$$A \to S$$

  we add a new initial symbol S' and rule

$$S' \to S$$

  This is called a *start-separated* grammar. No further reduction possible after reaching (ε, S', z)