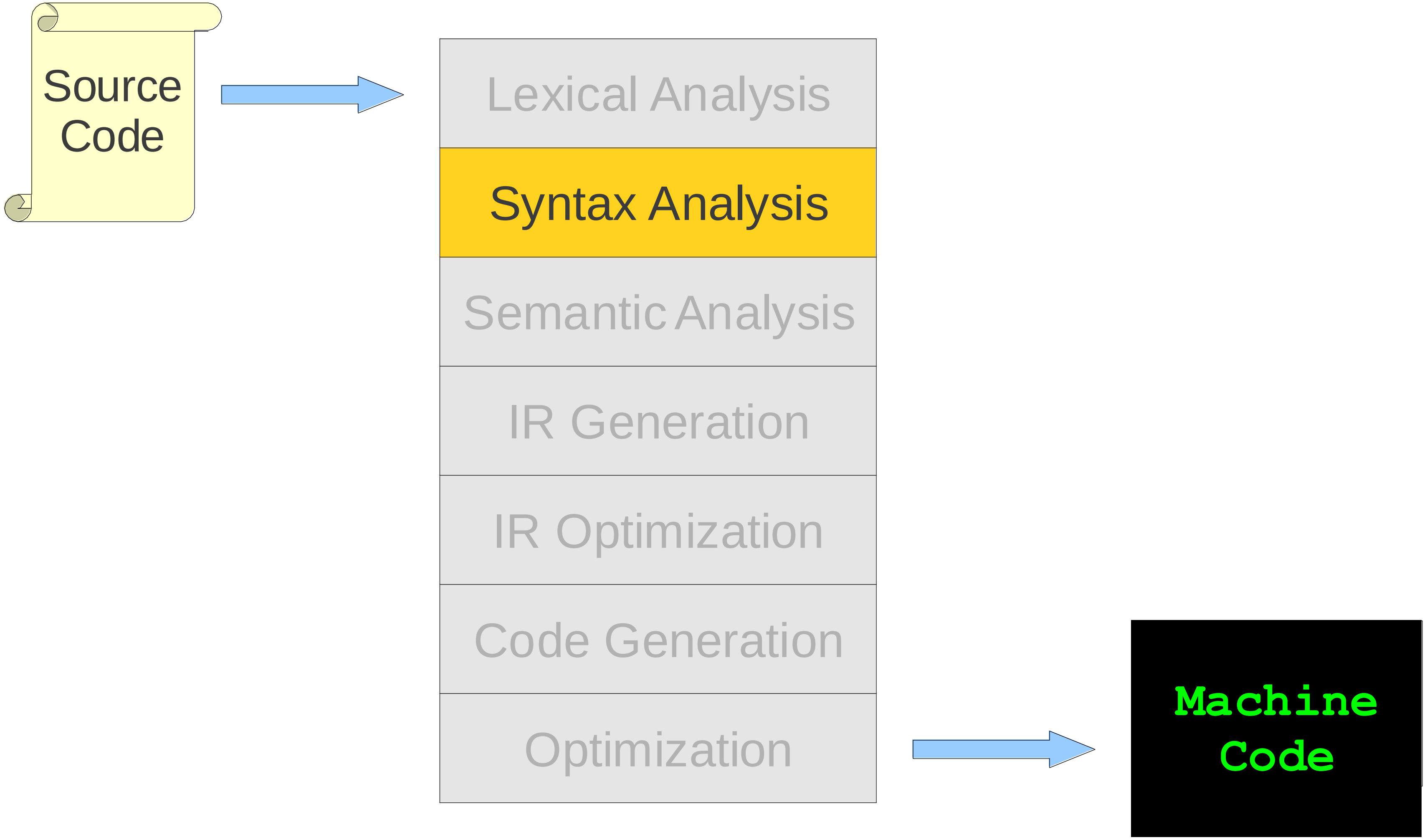
Compiler
– 3-3. Push Down Automata –

JIEUNG KIM

jieungkim@yonsei.ac.kr

YONSEI UNIVERSITY

Where are we?


Source Code
Lexical Analysis
Syntax Analysis
Semantic Analysis
IR Generation
IR Optimization
Code Generation
Optimization
Machine Code

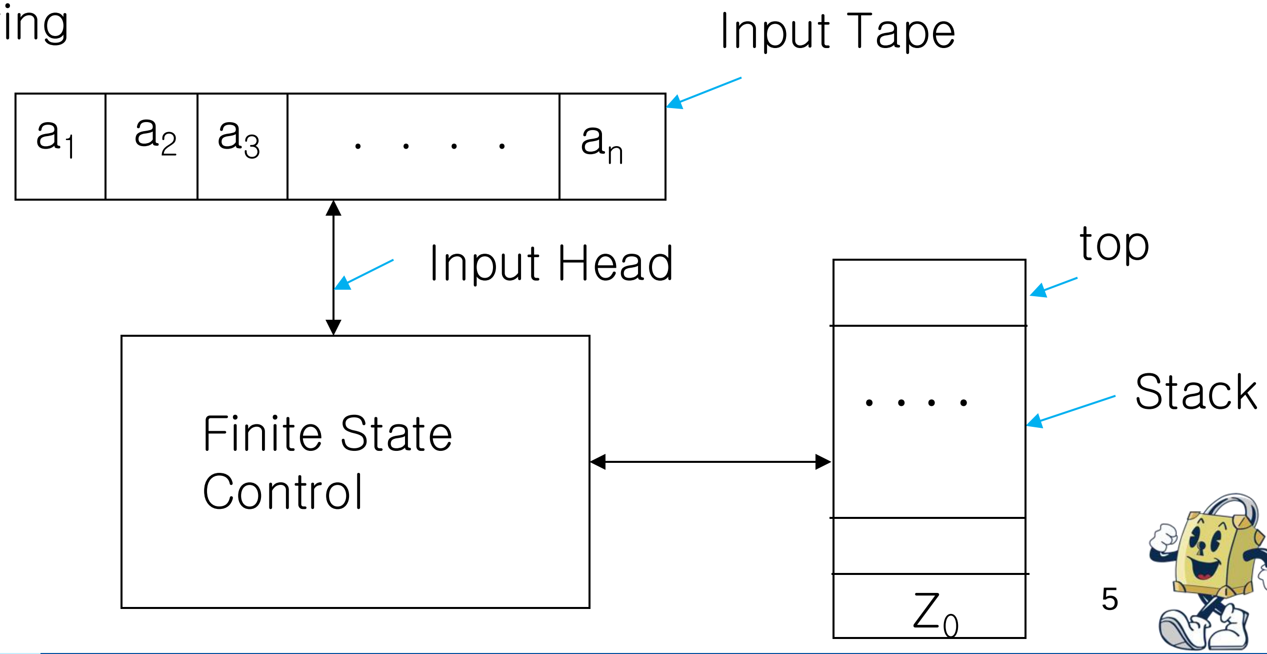
Outlines

- Role of the syntax analysis (parser)
- Context free grammar
- Push down automata
- Top-down parsing
- Bottom-up parsing
- Simple LR
- More powerful LR parsers and other issues in parsers
- Syntactic error handler
- Parser generator

Pushdown automata (PDA)

Pushdown automata (PDA)

- Pushdown automata (PDA)
  - A Pushdown Automata (PDA) is **ε-NFA** with a **stack**
  - On a transition, PDA
    - Consumes an input symbol (or stays by ε-move)
    - Goes to a new state (or stays in the old state)
    - Replaces the top of stack by any string
      - Does nothing
      - Pops the stack, or
      - Pushes a string onto the stack



Pushdown automata (PDA)

- Formal definition
  - PDA M = (Q, Σ, Γ, δ, q_0, Z_0, F) where

    Q: a finite set of states

    Σ: a finite set of input symbols

    Γ: a finite set of stack symbols

    q_0 (⊆ Q): a start state

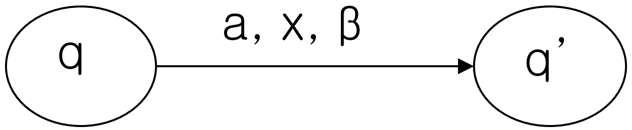
    Z_0 (⊆ Γ): stack start symbol

    F (⊆ Q): a finite set of final states

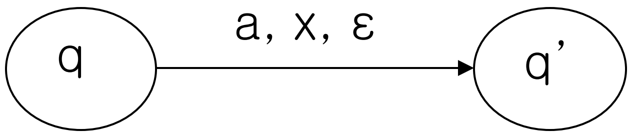
    δ: transition function is defined as: Q X (Σ ∪{ε}) X Γ → finite subsets of Q X Γ*

Pushdown automata (PDA)

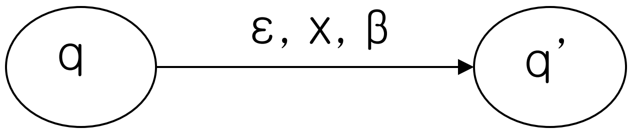
- Transition functions
  - Transition Function $\delta(q, a, x) = (q', \beta)$ where $q, q' \in Q$, $a \in \Sigma$, $x \in \Gamma$, $\beta \in \Gamma^*$
  - Case 1: If $a \neq \varepsilon$ and $\beta \neq \varepsilon$, then replace x by β.



  - Case 2: If $a \neq \varepsilon$ and $\beta = \varepsilon$, then x is popped.

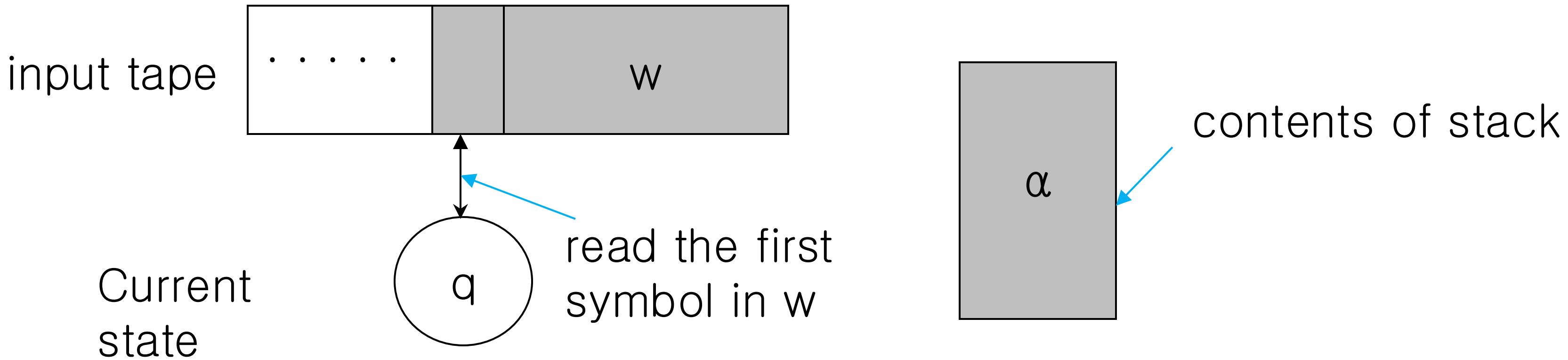


  - Case 3: If $a = \varepsilon$, then ε-move. (= head stays)



Pushdown automata (PDA)

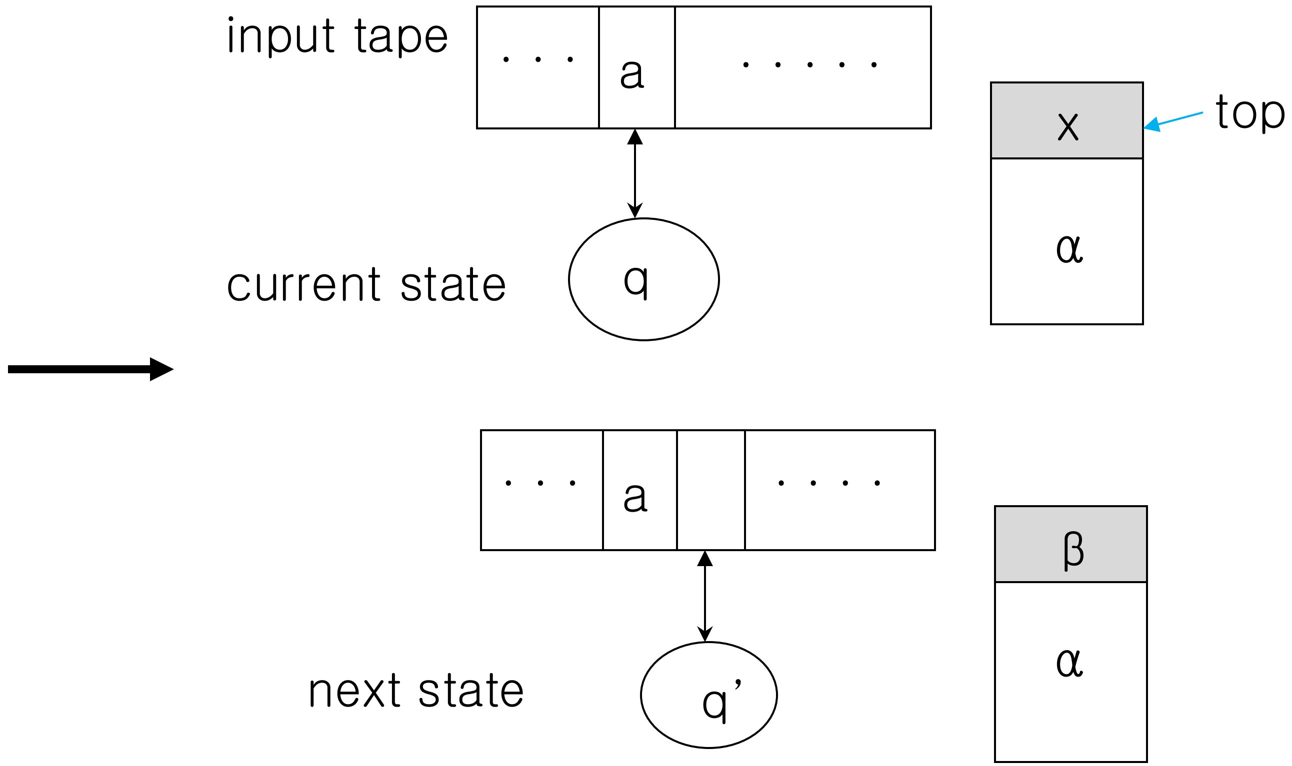
- Configuration
  - "Configuration" is defined as follows: (q, w, α) where $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$



  - If $w = \varepsilon$, then (q, ε, α): read all w
  - If $\alpha = \varepsilon$, then (q, w, ε): stack is empty
  - If $w = \varepsilon$ and $\alpha = \varepsilon$, then $(q, \varepsilon, \varepsilon)$: read all w, and stack is empty

Pushdown automata (PDA)

- Move (⊢)
  - Move (⊢) is defined as follows:
    - If $\delta(q, a, x) = (q', \beta)$, then $(q, aw, x\alpha) \vdash (q', w, \beta\alpha)$ where $a \in \Sigma$, $w \in \Sigma^*$, $x \in \Gamma$, α, $\beta \in \Gamma^*$

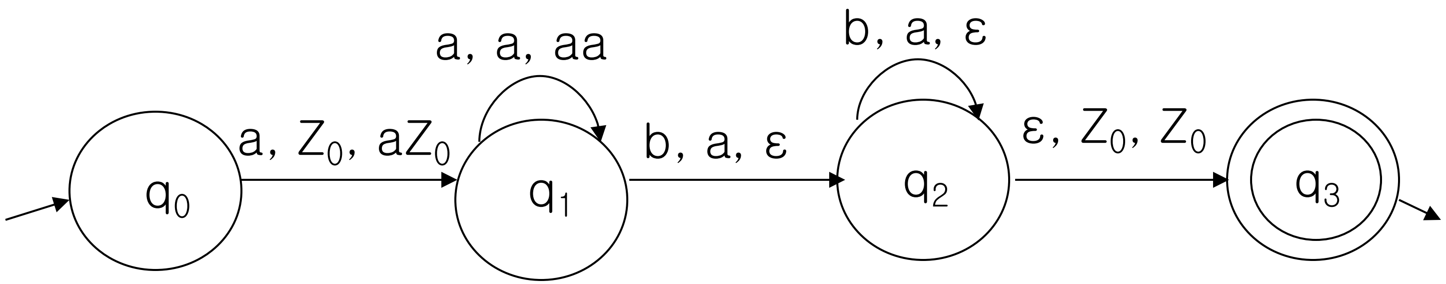


Pushdown automata (PDA)

- Example
  - Consider the following PDA M

M = ({q_0, q_1, q_2, q_3}, {a, b}, {a}, δ, q_0, Z_0, {q_3})
$\delta(q_0, a, Z_0) = (q_1, a\,Z_0)$: Push a
$\delta(q_1, a, a) = (q_1, aa)$: Push a's
$\delta(q_1, b, a) = (q_2, \varepsilon)$: See first b and Pop a
$\delta(q_2, b, a) = (q_2, \varepsilon)$: Pop
$\delta(q_2, \varepsilon, Z_0) = (q_3, Z_0)$: Go to final state

- Consider w = aabb
  - $(q_0, aabb, Z_0) \vdash (q_1, abb, aZ_0) \vdash (q_1, bb, aaZ_0) \vdash (q_2, b, aZ_0) \vdash (q_2, \varepsilon, Z_0) \vdash (q_f, \varepsilon, Z_0)$
  - w = aabb is accepted
- Consider w = abb
  - $(q_0, abb, Z_0) \vdash (q_1, bb, aZ_0) \vdash (q_2, b, Z_0)$
  - w = abb is rejected
- Consider w = aab
  - $(q_0, aab, Z_0) \vdash (q_1, ab, aZ_0) \vdash (q_1, b, aaZ_0) \vdash (q_2, \varepsilon, aZ_0)$
  - w = aab is rejected



Pushdown automata (PDA)

- Deterministic PDA (DPDA)
  - PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is **deterministic** if
    - $\delta(q, a, x)$ has at most one move for any $q \in Q$
      - (i.e., $\delta(q, a, x)$ is empty or has only one move)
    - If $\delta(q, a, x)$ is not empty for some $a \in \Sigma$, then $\delta(q, \varepsilon, x)$ must be empty
  - Example
    - $\delta(q, a, x) = (p, aa)$, $\delta(q, \varepsilon, x) = (r, aa)$ is not DPDA

Pushdown automata (PDA)

- Example – Deterministic PDA (DPDA)
  - L = { wcw^R | $w \in$ {a, b}* } (where c is a center mark)

$$M = (\{q_0, q_1, q_f\}, \{a, b, c\}, \{a, b\}, \delta, q_0, Z_0, \{q_f\})$$

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, b, b) = (q_0, bb)$

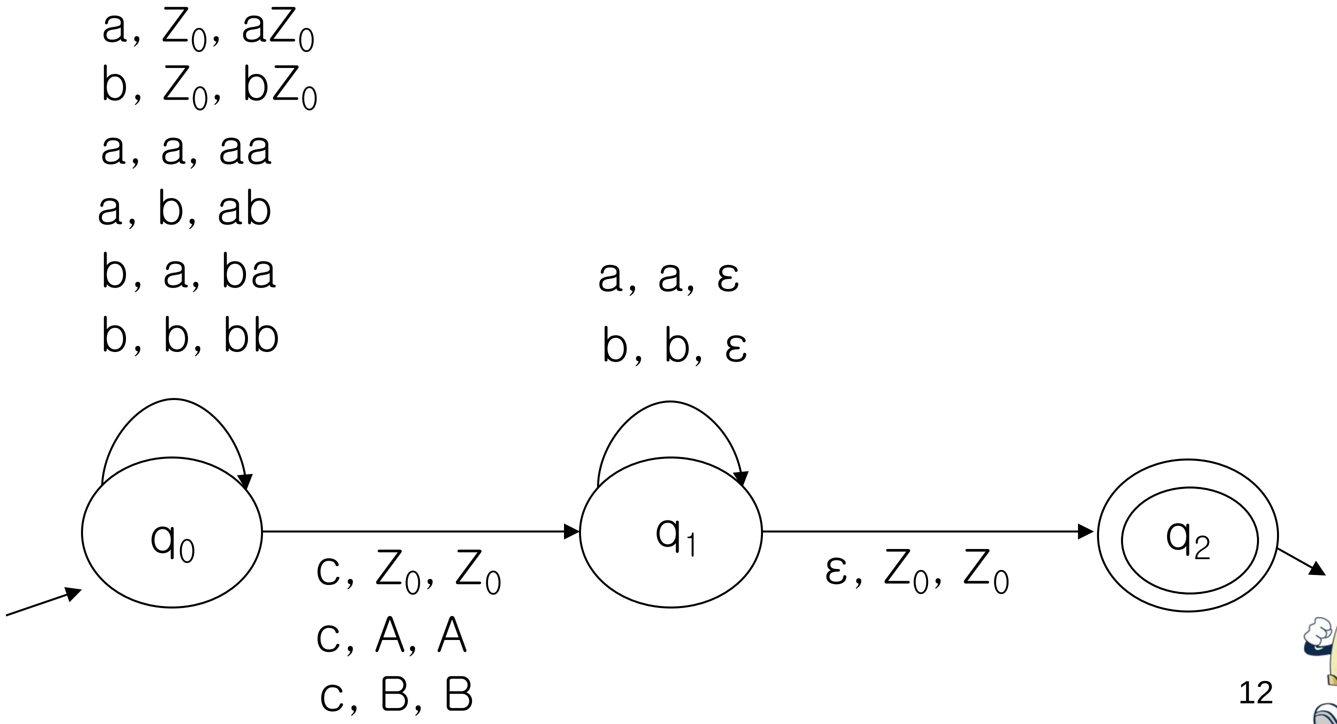
Push a's or b's

$\delta(q_0, c, a) = (q_1, a)$
$\delta(q_0, c, b) = (q_1, b)$
$\delta(q_0, c, Z_0) = (q_1, Z_0)$

Change state

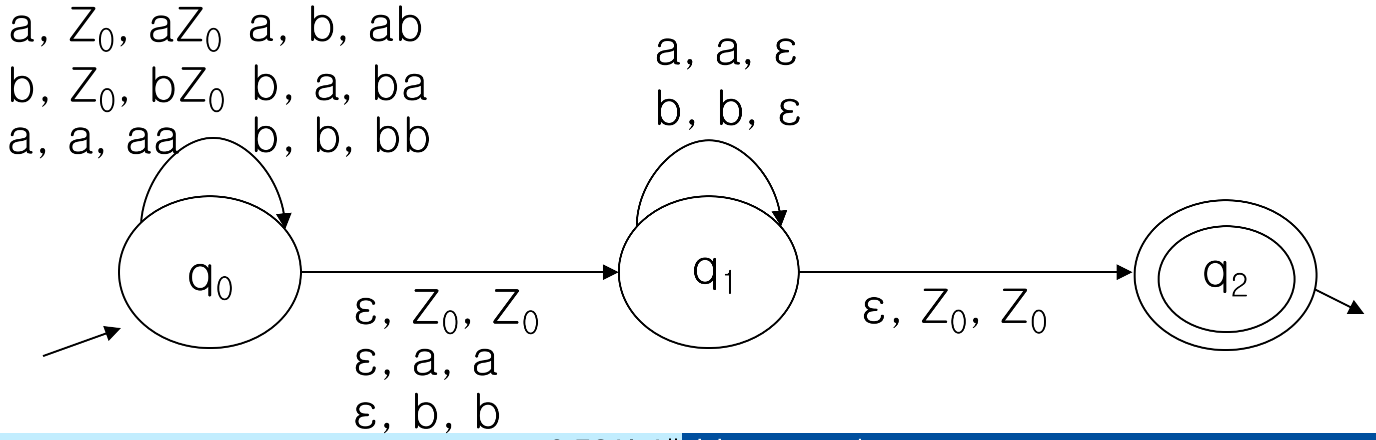
$\delta(q_1, a, a) = (q_1, \varepsilon)$
$\delta(q_1, b, b) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, Z_0) = (q_f, Z_0)$

Match and pop



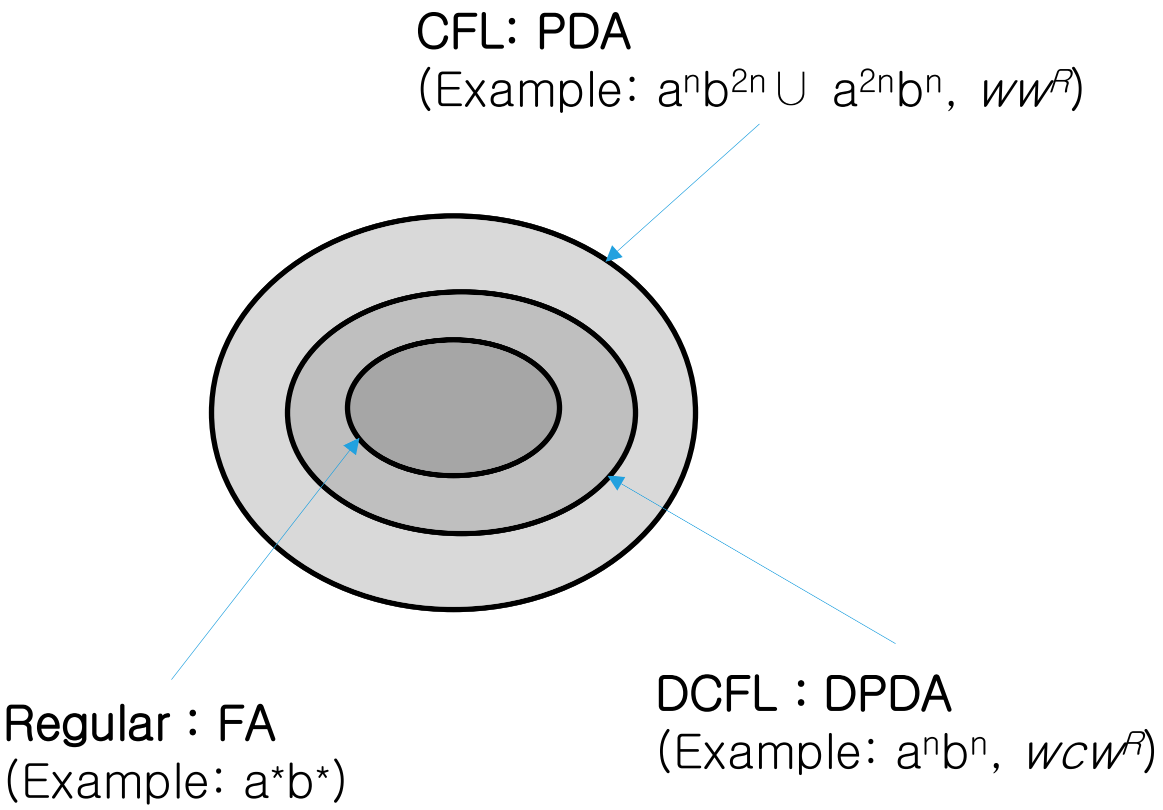
Pushdown automata (PDA)

- Nondeterministic PDA
  - Consider L = { $w \in \{a, b\}^*$ | ww^R}
    - Guess that you are reading *w*, then stay in state 0 and push the input symbols onto the stack
    - Guess that you are in the middle of ww^R, then go to state 1
    - You are now reading the head of ww^R, then compare it to the top of the stack. If they match, pop the stack, and remain in state 1, and then go to sleep If they don't match
    - If the stack is empty, then go to state 2, and accept



Pushdown automata (PDA)

- NPDA vs DPDA
  - A language accepted by Nondeterministic PDA is called CFL
    - Examples: $L = \{ww^R\}$, $L = \{a^n b^{2n}\} \cup \{a^{2n} b^n\}$
  - A language accepted by DPDA is called Deterministic CFL (DCFL)
    - Examples: $L = \{wcw^R\}$, $L = \{a^n b^{2n}\}$
  - PDA is inherently nondeterministic, and NPDA is more powerful than DPDA
  - There are CFL's that can not be accepted by DPDA
    - Examples: $L = \{ww^R\}$, $L = \{a^n b^{2n}\} \cup \{a^{2n} b^n\}$



Regular ⊂ DCFL ⊂ CFL

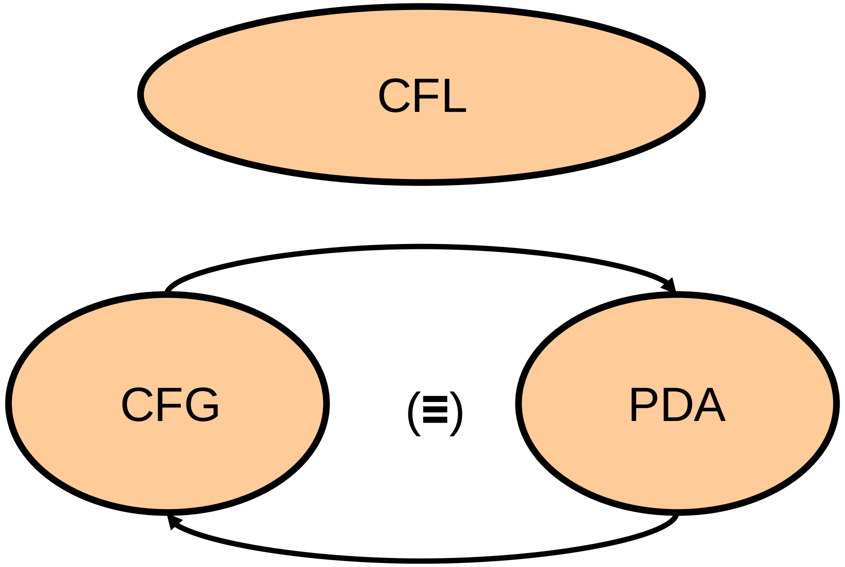
Pushdown automata (PDA)

- Power of DPDA
  - Most parsers (in practice) work as DPDA
    - Most programming languages can be described by DCFL
    - There are useful deterministic CFG's
    - Simple (*s*), *LL*, *LR* grammars
  - Every language accepted by DPDA has an unambiguous CFG
    - I.e., Every DCFL is unambiguous

YONSEI UNIVERSITY

Pushdown automata (PDA)

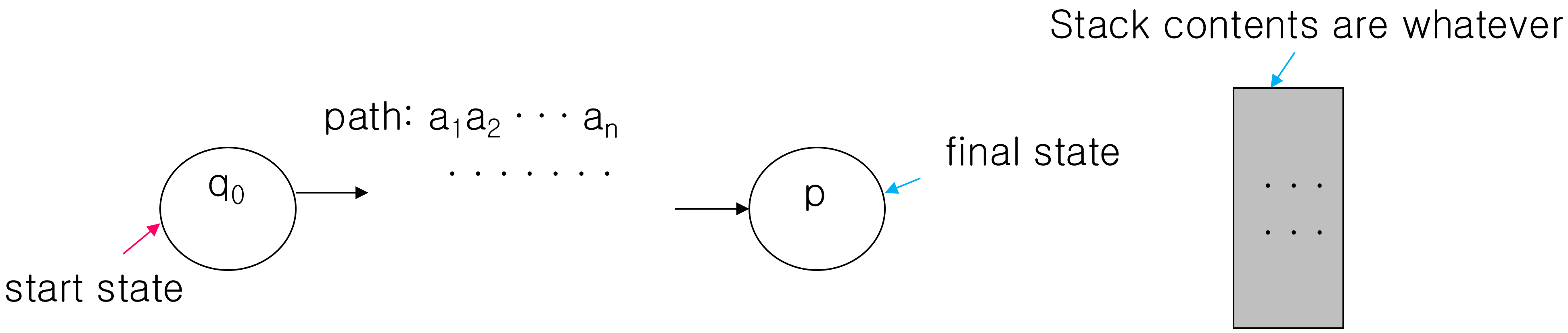
- Equivalence of PDA and CFG
  - A language is generated by a CFG iff it is accepted by a PDA



Questions?

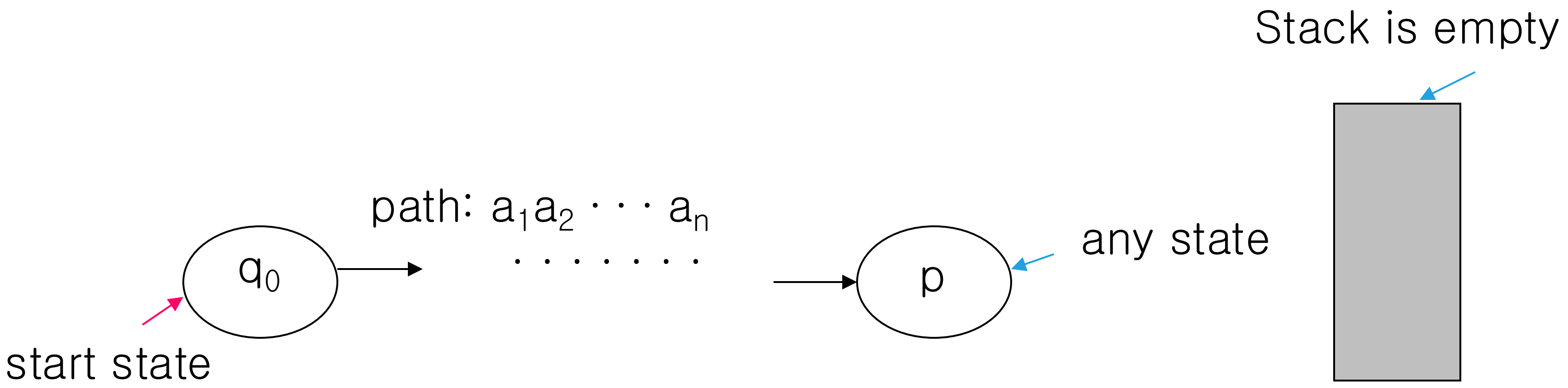
Pushdown automata (PDA) – supplementary page

- **PDA accepted by a final state**
  - An PDA M_f accepts a string $w = a_1a_2 \cdots a_n$ by a *final state* if there is path that
    - Begins at a start state
    - Ends at a final state
    - Has a sequence of labels $a_1a_2 \cdots a_n$



Pushdown automata (PDA) – supplementary page

- **PDA accepted by an empty stack**
  - An PDA M_ε accepts a string $w = a_1a_2 \cdots a_n$ by an *empty stack* if there is path that
    - Begins at a start state
    - Ends at any state
    - Has a sequence of labels $a_1a_2 \cdots a_n$
    - Stack must be empty finally



Pushdown automata (PDA) – supplementary page

- Context free language
  - A language accepted by PDA M_f by *final state*
    - $L(M_f) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash (p, \varepsilon, \alpha)$ for $p \in F\}$
    - $L(M_f)$ is called "Context Free Language"
    - Example: $L(M) = \{a^n b^n \mid n \geq 1\}$
  - A language accepted by PDA M_ε by *empty stack*
    - $L(M_\varepsilon) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash (p, \varepsilon, \varepsilon)$ for $p \in Q\}$
    - $L(M_\varepsilon)$ is also called "Context Free Language"
  - PDA M_f = PDA M_ε

Pushdown automata (PDA) – supplementary page

- Examples: construct PDA for the following languages:
  - L = $\{a^{2n}b^n \mid n \geq 0\}$
  - L = $\{a^n b^{2n} \mid n \geq 0\}$
  - L = $\{a^m b^n \mid m > n,\ m, n \geq 0\}$
  - L = $\{a^m b^n \mid m \neq n,\ m, n \geq 0\}$
  - L = $\{a^m b^m c^n \mid m, n \geq 0\}$
  - L = $\{a^{m+n} b^m c^n \mid m, n \geq 0\}$
  - L = $\{a^m b^n c^n d^m \mid m, n \geq 0\}$
  - L = $\{w \in \{a, b\}^* \mid w$ is not a palindrome.$\}$
  - L = $\{w \in \{a, b\}^* \mid \mathrm{NUMBER}_a(w) = \mathrm{NUMBER}_b(w)\}$

Pushdown automata (PDA) – supplementary page

- Convert CFG into PDA
  - Given CFG G, we construct a PDA that **simulates** ***leftmost*** derivations
  - Let $xA\beta \Rightarrow x\alpha\beta$. Then, PDA
    - Consumes input x by placing $A\beta$ on the stack, and then,
    - It pops A and pushes α by **ε-move**
  - PDA goes non-deterministically from $(q, y, A\beta)$ to $(q, y, \alpha\beta)$
  - Let G = (V, T, S, P) be a CFG, then we construct PDA M = ({q}, T, T ∪ V, δ, S, q) as follows:
    - If $A \rightarrow \alpha \in P$, then $\delta(q, \varepsilon, A) = (q, \alpha)$ (= Replace A by α in stack by ε-move.)
    - $\delta(q, X, X) = (q, \varepsilon)$ for all $X \in T$ (= If top symbol X is matched with input symbol X, erase it)
  - $L(G) = \{w \mid S \Rightarrow w \text{ and } w \in T^*\}$ iff $L(M) = \{w \in \Sigma^* \mid (q, w, S) \vdash (q, \varepsilon, \varepsilon)\}$

Pushdown automata (PDA) – supplementary page

- Example – convert CFG into PDA

$w = a + a * a$

CFG $G = (\{E, T, F\}, \{*, +, (,), a\}, E, P)$

$P: E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

Leftmost Derivation

$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T$

$\Rightarrow a + T * F \Rightarrow a + F * F \Rightarrow a + a * a$

PMA $M = (\{q\}, \{*, +, (,), a\}, V \cup T, \delta, E, q)$

$\delta(q, \varepsilon, E) = \{(q, E + T), (q, T)\}$

$\delta(q, \varepsilon, T) = (q, T * F), (q, F)\}$

$\delta(q, \varepsilon, F) = (q, (E)), (q, a)\}$

$\delta(q, X, X) = \{(q, \varepsilon)\}$ for all $X \in T$

PDA M: Simulate leftmost derivation.

$(q, a{+}a{*}a, E) \vdash (q, a{+}a{*}a, E{+}T) \vdash (q, a{+}a{*}a, T{+}T)$

$\vdash (q, a{+}a{*}a, F{+}T) \vdash (q, a{+}a{*}a, a{+}T)$

$\vdash (q, {+}a{*}a, {+}T) \vdash (q, a{*}a, T) \vdash (q, a{*}a, T{*}F)$

$\vdash (q, a{*}a, F{*}F) \vdash (q, a{*}a, a{*}F) \vdash (q, {*}a, {*}F)$

$\vdash (q, a, F) \vdash (q, a, a) \vdash (q, \varepsilon, \varepsilon)$

Questions?