Lecture 17 – Deterministic Pushdown Automata (DPDA)

COSE215: Theory of Computation

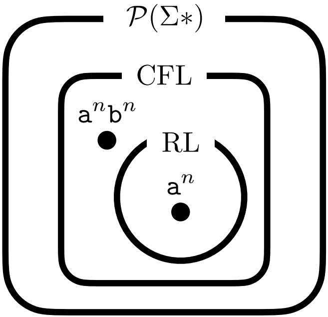
Jihyeok Park

PLRG

2024 Spring

Recall

- A **pushdown automaton (PDA)** is an extension of ϵ-NFA with a **stack**. Thus, PDA is **non-deterministic**.
  - Acceptance by **final states**
  - Acceptance by **empty stacks**
- Then, how about **deterministic PDA (DPDA)**?
- What is the **language class** of DPDA? Still, CFL?



PDA_{FS}
(by final states)

=

PDA_{ES}
(by empty stacks)

=

CFG

Contents

Contents

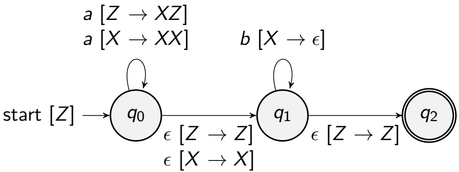
Definition (Deterministic Pushdown Automata (DPDA))

A **deterministic pushdown automaton (DPDA)** is a pushdown automaton having **at most one** one-step move ($\vdash$) from any configuration.

We can check it with the following conditions:

1. $|\delta(q, a, X)| \leq 1$ for all $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$, and $X \in \Gamma$.
2. If $\delta(q, \epsilon, X) \neq \varnothing$, then $\delta(q, a, X) = \varnothing$ for all $a \in \Sigma$.

For example, is the following PDA deterministic?



No, because it has multiple transitions for (q_0, a, Z).

Definition of Deterministic Pushdown Automata

PLRG

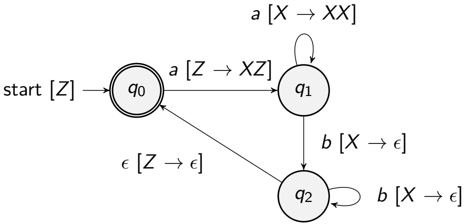
Definition (Deterministic Pushdown Automata (DPDA))

A **deterministic pushdown automaton (DPDA)** is a pushdown automaton having **at most one** one-step move ($\vdash$) from any configuration.

We can check it with the following conditions:

1. $|\delta(q, a, X)| \leq 1$ for all $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$, and $X \in \Gamma$.
2. If $\delta(q, \epsilon, X) \neq \varnothing$, then $\delta(q, a, X) = \varnothing$ for all $a \in \Sigma$.

However, the following PDA is **deterministic**:



$$
\begin{aligned}
(q_0, \mathtt{aabb}, Z) &\vdash (q_1, \mathtt{abb}, XZ) \\
&\vdash (q_1, \mathtt{bb}, XXZ) \\
&\vdash (q_2, \mathtt{b}, XZ) \\
&\vdash (q_2, \epsilon, Z) \\
&\vdash (q_0, \epsilon, \epsilon)
\end{aligned}
$$

Contents

PLRG

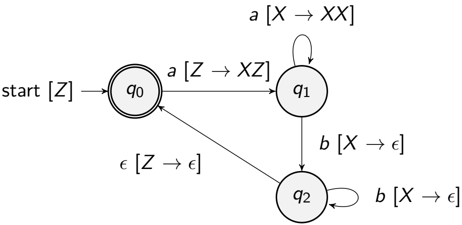
Definition (Deterministic Context-Free Languages (DCFLs))

A language L is a **deterministic context-free language (DCFL)** if and only if there exists a DPDA P such that $L = L_F(P)$ where $L_F(P)$ is the language accepted by **final states** of P.

For example, the following language is a DCFL:

$$L = \{a^n b^n \mid n \geq 0\}$$

because it is accepted by **final states** of the following **DPDA**:



Fact 1: DCFL $\subsetneq$ CFL

Fact 1: DCFL $\subsetneq$ CFL

① All DCFLs are CFLs **BUT** ② there exists a CFL that is not a DCFL.

1. DCFL $\subseteq$ CFL: By definition of DCFL, we can easily prove it.
2. CFL $\setminus$ DCFL $\neq \varnothing$: What is an example of a CFL that is not a DCFL?

   The following language is a CFL but not a DCFL:

$$L = \{ww^R \mid w \in \{a, b\}^*\} \in \text{CFL} \setminus \text{DCFL}$$

   The formal proof is complex, but we can intuitively understand it with the following two example words in L:

   - $ww^R = \texttt{abba} \in L$ where $w = \texttt{ab}$
   - $ww^R = \texttt{abbbba} \in L$ where $w = \texttt{abb}$

   When we read `b` after `ab`, we need to consider two possible actions: ① pop Y for `b` (for `abba`) or ② push Y for `b` (for `abbbba`).

PLRG

Fact 2: RL $\subsetneq$ DCFL

① All RLs are DCFLs **BUT** ② there exists a DCFL that is not an RL.

❶ RL $\subseteq$ DCFL: For a given RL L, consider its corresponding DFA D:

$$D = (Q, \Sigma, \delta, q_0, F)$$

Then, we can construct a DPDA P that accepts L as follows:

$$P = (Q, \Sigma, \{Z\}, \delta_P, q_0, Z, F)$$

where $\forall q \in Q.\ \forall a \in \Sigma.\ \delta_P(q, a, Z) = \{(\delta(q, a), Z)\}$ because

$$(q_0, w, Z) \vdash^* (q, \epsilon, Z) \quad \iff \quad \delta^*(q_0, w) = q \qquad \square$$

❷ DCFL $\setminus$ RL $\neq \varnothing$: We already know that the following language is a DCFL but not an RL:

$$L = \{a^n b^n \mid n \geq 0\} \in \text{DCFL} \setminus \text{RL}$$

Contents

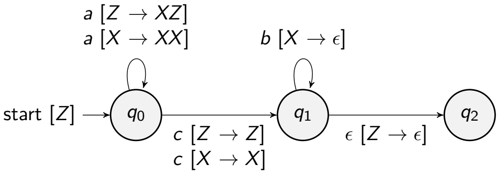
Definition ($DCFL_{ES}$)

A language L is a **deterministic context-free language by empty stacks ($DCFL_{ES}$)** if and only if there exists a DPDA P such that $L = L_E(P)$ where $L_E(P)$ is the language accepted by **empty stacks** of P.

For example, the following language is a $DCFL_{ES}$:

$$L = \{a^n c b^n \mid n \geq 0\}$$

because it is accepted by empty stacks of the following DPDA:

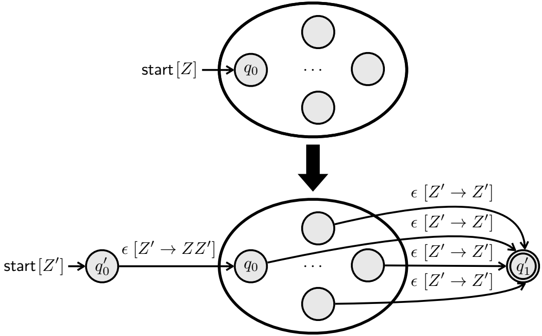


Fact 3: $\text{DCFL}_{\text{ES}} \subsetneq \text{DCFL}$

① All DCFL_{ES}s are DCFLs **BUT** ② there is a DCFL but not a DCFL_{ES}.

❶ $\boxed{\text{DCFL}_{\text{ES}} \subseteq \text{DCFL}}$: For a given DCFL_{ES} L, consider its corresponding DPDA P that accepts L by **empty stacks**.

Then, we can construct a DPDA P' that accepts L by **final states** as:



Fact 3: $\text{DCFL}_{\text{ES}} \subsetneq \text{DCFL}$

Fact 3: $\text{DCFL}_{\text{ES}} \subsetneq \text{DCFL}$

① All DCFL_{ES}s are DCFLs **BUT** ② there is a DCFL but not a DCFL_{ES}.

2. $\boxed{\text{DCFL} \setminus \text{DCFL}_{\text{ES}} \neq \varnothing}$: The following is a DCFL but not a DCFL_{ES}:

$$L = \{a^n b^n \mid n \geq 0\} \in \text{DCFL} \setminus \text{DCFL}_{\text{ES}}$$

Why?

The DPDA needs to accept the following two words by empty stacks:

- $w = \epsilon \in L$
- $w = \texttt{ab} \in L$

However, if a DPDA accepts the ϵ by empty stacks, then the stack must be empty at the beginning.

Thus, the PDA **cannot accept** $\texttt{ab}$ by empty stacks.

We can generalize it as **prefix property** of DCFL_{ES}.

Fact 4: DCFL_{ES} = DCFL having Prefix Property

PLRG

Definition (Prefix Property)

A language L has the **prefix property** if and only if for any word $w \in L$, any proper prefix of w is not in L:

$$\forall x, y \in \Sigma^*. ((xy \in L \land y \neq \epsilon) \implies x \notin L)$$

Fact 4: DCFL_{ES} = DCFL having Prefix Property

A language L is a DCFL_{ES} if and only if the language L is a DCFL having the **prefix property**.

For example, the following language is a **DCFL** but does **NOT** have the **prefix property** because $\epsilon \in L$ is a proper prefix of

$$L = \{a^n b^n \mid n \geq 0\}$$

Thus, L is a **DCFL** but **NOT** a $\textbf{DCFL}_{\textbf{ES}}$.

Fact 5: RL $\not\subset$ DCFL$_{\text{ES}}$

PLRG

Fact 5: RL $\not\subset$ DCFL$_{\text{ES}}$

There exists a RL that is not a DCFL$_{\text{ES}}$.

- $\boxed{\text{RL} \setminus \text{DCFL}_{\text{ES}} \neq \varnothing}$: For example, the following language is a **RL** but does **NOT** have the **prefix property**:

$$L = \{a^n \mid n \geq 0\} \in \text{RL} \setminus \text{DCFL}_{\text{ES}}$$

because aa $\in L$ is a proper prefix of aaaa $\in L$.

Thus, L is a **RL** but **NOT** a **DCFL$_{\text{ES}}$**.

Contents

Inherent Ambiguity of DCFLs

Definition (Inherent Ambiguity)

A language L is **inherently ambiguous** if all CFGs whose languages are L are ambiguous. (i.e. there is no unambiguous grammar for L)

What is the relationship of **inherently ambiguous languages** and DCFLs?

It satisfies the following fact:

$$\text{DCFL} \subsetneq \text{Non Inherently Ambiguous Languages}$$

We prove this fact by the following three steps:

1. $\boxed{\text{DCFL}_{\text{ES}} \subseteq \text{Non Inherently Ambiguous Languages}}$
2. $\boxed{\text{DCFL} \subseteq \text{Non Inherently Ambiguous Languages}}$ (using ①)
3. $\boxed{\text{Non Inherently Ambiguous Languages} \setminus \text{DCFL} \neq \varnothing}$

PLRG

Fact 6: DCFL $\subsetneq$ Non Inherently Ambiguous Languages

All DCFLs have at least one corresponding unambiguous grammars **BUT** there exists a non inherently ambiguous language that is not a DCFL.

1. A DCFL$_{ES}$ has an unambiguous grammar: For a given DCFL$_{ES}$ L and its corresponding DPDA P, we can define a CFG for P as follows:
   - For all $0 \leq j < n$,

     $$S \to A^{Z}_{0,j}$$

   - For all transition $(q_j, X_1 \cdots X_m) \in \delta(q_i, a, X)$ where $q_i \in Q$, $a \in \Sigma \cup \{\epsilon\}$, and $X \in \Gamma$ and combinations $0 \leq k_1, \cdots, k_m < n$:

     $$A^{X}_{i,k_m} \to a\ A^{X_1}_{j,k_1}\ A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m}$$

   For any word $w \in L$, w has a unique moves from the initial configuration to the final configuration in P. And, we know that:

   $$A^{X}_{i,j} \Rightarrow^* w \qquad \text{if and only if} \qquad (q_i, w, X) \vdash^* (q_j, \epsilon, \epsilon)$$

   Thus, the above CFG is **unambiguous**.

PLRG

Fact 6: DCFL $\subsetneq$ Non Inherently Ambiguous Languages

All DCFLs have at least one corresponding unambiguous grammars **BUT** there exists a non inherently ambiguous language that is not a DCFL.

2. A DCFL has an unambiguous grammar : For a given DCFL L, we can define another DCFL L' with a special symbol \$ as follows:

$$L' = L\$ = \{w\$ \mid w \in L\}$$

Then, L' is a DCFL_{ES} because it has the **prefix property**, and L' has an **unambiguous grammar** G'. Now, we can define an **unambiguous grammar** G for L by treating \$ as a variable with a rule $\$ \rightarrow \epsilon$.

For example, $L = \{a^n b^n \mid n \geq 0\}$ is DCFL, then $L' = \{a^n b^n \$ \mid n \geq 0\}$ is a DCFL_{ES} and its **unambiguous grammar** G' is:

$$S \rightarrow X\$ \qquad X \rightarrow \mathtt{a}X\mathtt{b} \mid \epsilon$$

Then, the **unambiguous grammar** G for L is:

$$S \rightarrow X\$ \qquad X \rightarrow \mathtt{a}X\mathtt{b} \mid \epsilon \qquad \$ \rightarrow \epsilon$$

Fact 6: DCFL $\subsetneq$ Non Inherently Ambiguous Languages

All DCFLs have at least one corresponding unambiguous grammars **BUT** there exists a non inherently ambiguous language that is not a DCFL.

3. $\boxed{\text{Non Inherently Ambiguous Languages} \setminus \text{DCFL} \neq \varnothing}$: The following language is a **non inherently ambiguous language** but **not** a **DCFL**:

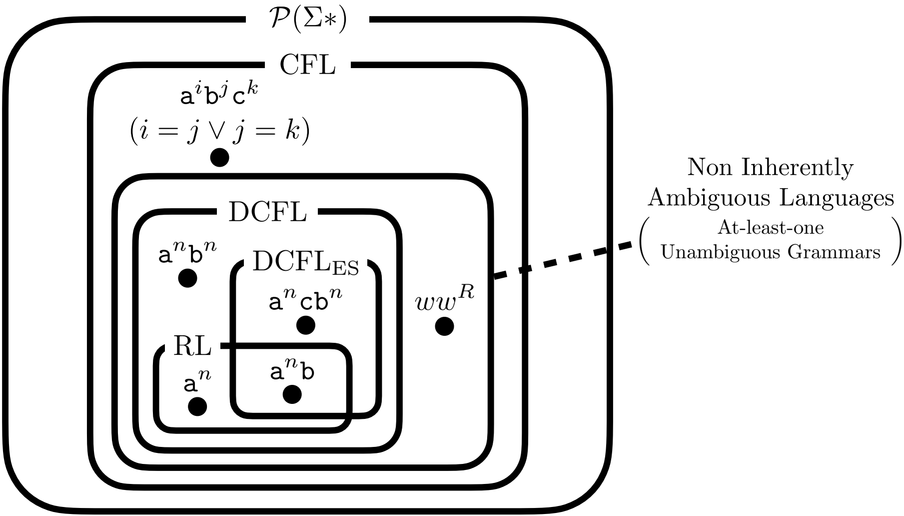
$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

because the following **unambiguous grammar** G represents L:

$$S \to aSa \mid bSb \mid \epsilon$$

but we already know that L is **not** a **DCFL**.

Summary



$\mathcal{P}(\Sigma*)$
CFL
$\texttt{a}^i\texttt{b}^j\texttt{c}^k$
$(i = j \lor j = k)$
DCFL
$\texttt{a}^n\texttt{b}^n$
DCFL_{ES}
$\texttt{a}^n\texttt{c}\texttt{b}^n$
RL
$\texttt{a}^n$
$\texttt{a}^n\texttt{b}$
ww^R
Non Inherently
Ambiguous Languages
(At-least-one
Unambiguous Grammars)

Next Lecture

- Normal Forms of Context-Free Grammars

Jihyeok Park
jihyeok_park@korea.ac.kr
https://plrg.korea.ac.kr