Lec 11. Properties of PDA and deterministic PDA

Eunjung Kim

Closure and non-closure properties of CFL

Context-free languages are closed under

- substitution
- union
- concatenation
- kleene star $(*)$ and positive star $(+)$
- reversal
- intersection with a regular language

and not closed under

- intersection
- complementation
- $L_1 - L_2$

Substitution

Given a CFL L over Σ and $a \in \Sigma$, we want to define a new language by substituting any occurrence of a by all strings of L_a.

Formal definition of substitution

For a finite alphabet Σ, let s be a mapping from Σ to the set of all languages, called a <u>subsitution</u> on Σ.

- For a string $w = a_1, \ldots, a_n \in \Sigma^*$, $s(w)$ is defined as

$$s(a_1) \cdot s(a_2) \cdot \cdots \cdot s(a_n).$$

- For a language L over Σ, $s(L)$ is defined as

$$\bigcup_{w \in L} s(w).$$

Substitution: example

- Let s be a substitution on $\Sigma = \{0, 1\}$ with $s(0) = \{a^n b^n : n \geq 1\}$ and $s(1) = \{aa, bb\}$.
- Let $L = \{0^i : i \geq 0\}$. Then $s(L)$ is the set of all strings of the form

??????????????????????

CFL's closed under substitution

Theorem

If L is a CFL and $s(a)$ is a CFL for each $a \in \Sigma$, then $s(L)$ is a CFL.

Union

Concatenation

Kleene and positive closure

Reversal

Intersection with a regular language

Theorem

If L is a CFL and R is a regular language, then $L \cap R$ is a CFL.

Deterministic Pushdown Automata

An attempt to eliminate the nondeterminism in a PDA using a similar trick as for NFA is doomed as ϵ-move (i.e. reading an input symbol without popping, or pushing, or the other way around) is an essential feature of PDA. Nonetheless, one can limit the possible move to an unique move: at state q, any input symbol a and a stack symbol u at hand, there are four possible moves.

- read, pop
- read, no pop
- no read, pop
- no read, no pop

Deterministic Pushdown Automata

Deterministic PDA

A deterministic pushdown automaton is a PDA $(Q, \Sigma, \Gamma, \delta, q_0, F)$ such that the transition function δ is a mapping

$$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow (Q, \Gamma_\epsilon) \times \{\emptyset\}$$

which additionally satisfies the following condition:

> *for every $q \in Q, a \in \Sigma$ and $x \in \Gamma$, exactly one of the transition*
> $$\delta(q, a, x), \delta(q, a, \epsilon), \delta(q, \epsilon, x) \text{ and } \delta(q, \epsilon, \epsilon)$$
> *is not $\emptyset$.*

The language recognized by some deterministic PDA is called a deterministic context-free language.

DETERMINISTIC OR NOT?

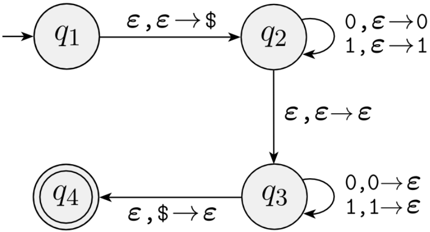

Figure 2.19, Sipser 2012

Example of deterministic CFL

The language $\{0^n 1^n : n \geq 0\}$ is a deterministic CFL.

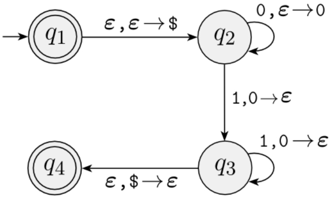

Figure 2.15, Sipser 2012

Power and limitation

- Regular languages $\subsetneq$ Deterministic CFLs $\subsetneq$ CFL.
- Consider $L_1 = \{0^n 1^n : n \geq 0\}$.
- Consider $L_2 = \{ww^R : w \in \{0,1\}^*\}$.
  1. Both 0110 and 011110 are in L.
  2. After reading 01, upon reading the third 1, PDA needs to (deterministically) choose between "matching" (by pop move) and "yet reading w" (by push move).
  3. $\{wcw^R : w \in \{0,1\}^*\}$, where c is a "middle place marker", is deterministic CFL.
- Deterministic CFLs $\subsetneq$ non inherently ambiguous languages
- Consider the grammar $S \to 0S0 \quad |1S1 \quad |\epsilon$ for L_2.

FACTS

- A is a deterministic CFL if and only if $A \dashv$ is a deterministic CFL, where $\dashv$ is a 'endmark', i.e. a special symbol to signify the end of an input string.
- The latter type of language is called an endmarked language.
- Language generated by a deterministic CFG $\subsetneq$ deterministic CFLs.
- An endmarked language is generated by a deterministic CFG if and only if it is deterministic context-free (i.e. recognized by a deterministic PDA).