CS 133 : Automata Theory and Computability

Lecture Slides

Context-Free Grammars and Languages

Francis George C. Cabarle

Algorithms and Complexity Laboratory
Department of Computer Science
University of the Philippines Diliman
fccabarle@up.edu.ph

Day 14

Context-Free Grammars and Languages

Deterministic Pushdown Automata

Non-Context Free Languages

Context-Free Grammars and Languages

Deterministic Pushdown Automata

Non-Context Free Languages

DPDA

Definition

A pushdown automaton $(Q, \Sigma, \Gamma, \delta, q_0, F)$, is deterministic if and only if the following conditions are met:

1. $\delta(q, a, t)$ has at most one element for any $q \in Q$, $a \in \Sigma_\varepsilon$, and $t \in \Gamma$.
2. If $\delta(q, a, t)$ is nonempty for some $a \in \Sigma$, then $\delta(q, \varepsilon, t)$ must be empty.

DPDA

Definition

A pushdown automaton $(Q, \Sigma, \Gamma, \delta, q_0, F)$, is deterministic if and only if the following conditions are met:

1. $\delta(q, a, t)$ has at most one element for any $q \in Q$, $a \in \Sigma_\varepsilon$, and $t \in \Gamma$.
2. If $\delta(q, a, t)$ is nonempty for some $a \in \Sigma$, then $\delta(q, \varepsilon, t)$ must be empty.

i.e., a PDA is deterministic if it never has more than one choice of transiton for a given *state, input symbol* (including ε), and *stack symbol*.

DPDA

Definition

A pushdown automaton $(Q, \Sigma, \Gamma, \delta, q_0, F)$, is deterministic if and only if the following conditions are met:

1. $\delta(q, a, t)$ has at most one element for any $q \in Q$, $a \in \Sigma_\varepsilon$, and $t \in \Gamma$.
2. If $\delta(q, a, t)$ is nonempty for some $a \in \Sigma$, then $\delta(q, \varepsilon, t)$ must be empty.

i.e., a PDA is deterministic if it never has more than one choice of transiton for a given *state, input symbol* (including ε), and *stack symbol.*

e.g. DPDA for $L_{wwr} = \{ww^R \mid w \in \{0, 1\}^*\}$?

DPDA

Definition

A pushdown automaton $(Q, \Sigma, \Gamma, \delta, q_0, F)$, is deterministic if and only if the following conditions are met:

1. $\delta(q, a, t)$ has at most one element for any $q \in Q$, $a \in \Sigma_\varepsilon$, and $t \in \Gamma$.
2. If $\delta(q, a, t)$ is nonempty for some $a \in \Sigma$, then $\delta(q, \varepsilon, t)$ must be empty.

i.e., a PDA is deterministic if it never has more than one choice of transiton for a given *state, input symbol* (including ε), and *stack symbol*.

e.g. DPDA for $L_{wwr} = \{ww^R \mid w \in \{0,1\}^*\}$?
L_{wwr} has no DPDA. How about $L_{wcwr} = \{wcw^R \mid w \in \{0,1\}^*\}$?

The languages recognized by DPDA

The languages recognized by DPDA

- The two modes of acceptance, final state and empty stack, are not the same for DPDAs.
- All the regular languages are accepted (by final state) by DPDAs.

The languages recognized by DPDA

- The two modes of acceptance, final state and empty stack, are not the same for DPDAs.
- All the regular languages are accepted (by final state) by DPDAs. But there are nonregular languages recognized by DPDAs.

The languages recognized by DPDA

- The two modes of acceptance, final state and empty stack, are not the same for DPDAs.
- All the regular languages are accepted (by final state) by DPDAs. But there are nonregular languages recognized by DPDAs.
- The DPDA languages are context-free languages, and in fact are languages that have unambiguous CFGs.

The languages recognized by DPDA

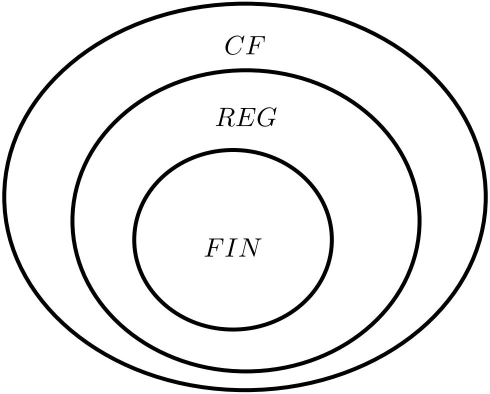
- The two modes of acceptance, final state and empty stack, are not the same for DPDAs.
- All the regular languages are accepted (by final state) by DPDAs. But there are nonregular languages recognized by DPDAs.
- The DPDA languages are context-free languages, and in fact are languages that have unambiguous CFGs.
- Thus, the DPDA languages lie strictly between the regular languages and the context-free languages.

Context-Free Grammars and Languages

Deterministic Pushdown Automata

Non-Context Free Languages

Chomsky hierarchy (so far)



(non)Context-free languages

$L_{ii} = \{0^i 1 0^i \mid i \geq 1\} \in CF.$

$L_{iii} = \{0^i 1 0^i 1 0^i \mid i \geq 1\} \notin CF$.

$\{ww^R \mid w \in \Sigma^*\} \in CF.$

$\{ww \mid w \in \Sigma^*\} \notin CF.$

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L,

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L, there is a $p \in \mathbb{N}$ such that

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L*, there is a* $p \in \mathbb{N}$ *such that for every* $s \in L$ *of length* $|s| \geq p$

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L, there is a $p \in \mathbb{N}$ such that for every $s \in L$ of length $|s| \geq p$ there exists u, v, w, x, y such that $s = uvwxy$ and:

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L*, there is a* $p \in \mathbb{N}$ *such that for every* $s \in L$ *of length* $|s| \geq p$ *there exists* u, v, w, x, y *such that* $s = uvwxy$ *and:*

1. $|vx| > 0$,

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L, there is a $p \in \mathbb{N}$ such that for every $s \in L$ of length $|s| \geq p$ there exists u, v, w, x, y such that $s = uvwxy$ and:

1. $|vx| > 0$,
2. $|vwx| \leq p$,

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L, there is a $p \in \mathbb{N}$ such that for every $s \in L$ of length $|s| \geq p$ there exists u, v, w, x, y such that $s = uvwxy$ and:

1. $|vx| > 0$,
2. $|vwx| \leq p$,
3. *for each* $i \geq 0, uv^iwx^iy \in L$.

Pumping Lemma for Context-Free Languages

Theorem

For every context-free language L, there is a $p \in \mathbb{N}$ such that for every $s \in L$ of length $|s| \geq p$ there exists u, v, w, x, y such that $s = uvwxy$ and:

1. $|vx| > 0$,
2. $|vwx| \leq p$,
3. *for each* $i \geq 0, uv^iwx^iy \in L$.

We use the pumping lemma to prove that a language L is not context-free.

Proof for the pumping lemma, part I

- Let $G = (N, T, P, S)$ be a CFG in CNF[1] and let $L(G) = A$ is a CFL. N.t.s. the statement and conditions of the lemma are satisfied.
- *Any* parse tree using G has a *node* with *at most 2 children* since G is in CNF.
  Observe: In any parse tree of G, at most 2^h leaves within h steps from S.
  If height of parse tree $\leq h$, length of generated string has length $\leq 2^h$, or if string generated has length $\geq 2^h + 1$, its parse tree must be at height $\geq h + 1$.
- Set $p = 2^{|N|+1}$ for N in G, so if $s \in A$ has length $\geq p$ then parse tree for s has height $\geq |N| + 1$ since $2^{|N|+1} \geq 2^{|N|} + 1$.
- Let τ be one of the parse trees for s where τ has the *least number of nodes* among all trees for s.
  Since τ has height $\geq |N| + 1$, select its *longest path* from S to a leaf with length $\geq |N| + 1$. This path has $\geq |N| + 2$ nodes: one from T and $|N| + 1$ from N.

Proof for the pumping lemma, part II

- G only has $|N|$ variables, so some $R \in N$ appears ≥ 2 on this longest path.
  Later we select R to appear in the lowest $|N| + 1$ variables on this path.
- Let $s = uvwxy$. In the tree τ, let the *first* and *second* appearances of R from S be subtrees generating vwx and w, respectively.
  Both subtrees are generated by the same variable R so we can *replace one for the other* and still have a *valid parse tree.*
  Obtain part of condition 3, i.e. uv^iwx^iy for $i \geq 1$, if we repeatedly replace the second appearance with the first appearance of R; we obtain $i = 0$ by replacing the the first with the second appearance of R.
- $vx \neq \varepsilon$, i.e. condition 1, since if v and x were both ε then replacing the first with the second appearance of R in the parse tree for s results in a tree with *fewer nodes* than τ: this is not possible based on how we selected τ previously.

Proof for the pumping lemma, part III

- For condition 2, i.e. $|vwx| \leq p$: in the parse tree for s the first appearance of R generates vwx.
  We selected R so both of its appearances are in the bottom $|N| + 1$ variables of the longest path in τ, so the subtree where R generates vwx is at height $\leq |N| + 1$.
  A tree of height $|N| + 1$ can generate strings of length $\leq 2^{|N|+1} = p$. QED.

[1] This proof is another variation from the proof by [Sipser 2005] and [Hopcroft et al 2001].

Some Non-Context Free Languages

Example

1. $B = \{a^n b^n c^n \mid n \geq 0\}$

Some Non-Context Free Languages

Example

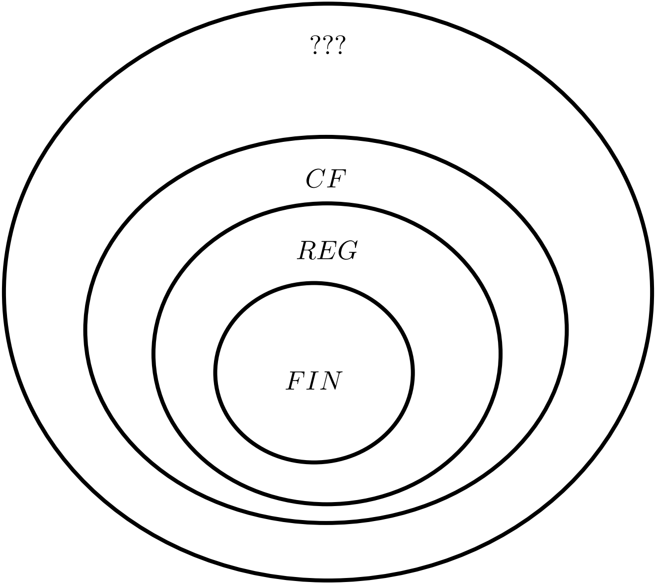
1. $B = \{a^n b^n c^n \mid n \geq 0\}$
2. $C = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$

Some Non-Context Free Languages

Example

1. $B = \{a^n b^n c^n \mid n \geq 0\}$
2. $C = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$
3. $D = \{ww \mid w \in \{0,1\}^*\}$

Chomsky hierarchy (now)



Fin (wakas)

Thanks for the attention.

Questions?

Read examples and answer exercises from:

[Sipser 2005] Chapter 2.3, or

[Hopcroft et al 2001] Chapter 7.2

References:

[Sipser 2005] M. Sipser. *Introduction to the Theory of Computation*: 2ed. PWS Publishing Company, 2005.

[Hopcroft, Ullman 1979] J. Hopcroft, J. Ullman. *Introduction to Automata Theory, Languages, and Computation*: Addison-Wesley, 1979.

[Hopcroft et al 2001] J. Hopcroft, R. Motwani, J. Ullman. *Introduction to Automata Theory, Languages, and Computation*: Addison-Wesley, 2001.