# BLM2502 Theory of Computation

Spring 2016

# BLM2502 Theory of Computation

- **Course Outline**

| Week | Content |
|---|---|
| 1 | Introduction to Course |
| 2 | Computability Theory, Complexity Theory, Automata Theory, Set Theory, Relations, Proofs, Pigeonhole Principle |
| 3 | Regular Expressions |
| 4 | Finite Automata |
| 5 | Deterministic and Nondeterministic Finite Automata |
| 6 | Epsilon Transition, Equivalence of Automata |
| 7 | Pumping Theorem |
| 8 | April 10 - 14 week is the first midterm week |
| 9 | Context Free Grammars |
| 10 | Parse Tree, Ambiguity, |
| **11** | **Pumping Theorem** |
| 12 | Turing Machines, Recognition and Computation, Church-Turing Hypothesis |
| 13 | Turing Machines, Recognition and Computation, Church-Turing Hypothesis |
| 14 | May 22 – 27 week is the second midterm week |
| 15 | Review |
| 16 | Final Exam date will be announced |

# The Pumping Lemma for CFL's

# Simplifications of Context-Free Grammars

# A Substitution Rule

$S \rightarrow \cancel{aB}$ | ab | aaA

$A \rightarrow aaA$ |

$A \rightarrow a\cancel{bB}c$ | abbc | abaAc

$\cancel{B \rightarrow aA}$

$\cancel{B \rightarrow b}$

Substitute

$B \rightarrow b$

Equivalent grammar

$S \rightarrow aB \mid ab$

$A \rightarrow aaA$

$A \rightarrow abBc \mid abbc$

$B \rightarrow aA$

$$S \to aB \mid ab$$

$$A \to aaA$$

$$A \to abBc \mid abbc$$

$$B \to aA$$

Substitute

$$B \to aA$$

$$S \to \cancel{aB} \mid ab \mid aaA$$

$$A \to aaA$$

$$A \to \cancel{abBc} \mid abbc \mid abaAc$$

Equivalent grammar

In general: $A \to xBz$

$B \to y_1$

Substitute $B \to y_1$

$A \to xBz \mid xy_1z$

equivalent grammar

# Nullable Variables

$\varepsilon-$production :

$$X \to \varepsilon$$

Nullable Variable:

$$Y \Rightarrow \ldots \Rightarrow \varepsilon$$

Example:

$$S \to aMb \mid ab$$

$$M \to aMb \mid ab$$

$$\cancel{M \to \varepsilon}$$

Nullable variable

$\varepsilon-$production

# Removing $\varepsilon$ – productions

$$S \to aMb \mid ab$$

$$M \to aMb \mid ab$$

$$\cancel{M \to \varepsilon}$$

Substitute

$$M \to \varepsilon$$

$$S \to aMb \mid ab$$

$$M \to aMb \mid ab$$

After we remove all the ε - productions
all the nullable variables disappear
(except for the start variable)

# Unit-Productions

Unit Production: $X \to Y$

(a single variable in both sides)

Example:

$$S \to aA$$

$$A \to a$$

$$\boxed{A \to B}$$

$$\boxed{B \to A}$$

Unit Productions

$$B \to bb$$

# Removal of unit productions:

$$S \rightarrow a\cancel{A} \;|\; a\cancel{B} \;|\; abb \;|\; aa \;|\; aa$$

$$\cancel{A \rightarrow a}$$

$$\cancel{A \rightarrow B}$$

$$\cancel{B \rightarrow A} \;|\; a$$

$$\cancel{B \rightarrow bb}$$

**Substitute**

$$A \rightarrow B$$

$$S \rightarrow aA \,|\, aB$$

$$A \rightarrow a$$

$$B \rightarrow A \,|\, B$$

$$B \rightarrow bb$$

Unit productions of form $X \to X$ can be removed immediately

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to A \mid \cancel{B}$$
$$B \to bb$$

Remove $B \to B$

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to A$$
$$B \to bb$$

$$S \to aA \mid aB$$

$$A \to a$$

$$\cancel{B \to A}$$

$$B \to bb$$

→ B gördüğüm yere A yazarım

**Substitute**

$$B \to A$$

$$S \to \cancel{aA} \mid aB \mid aA$$

$$A \to a$$

$$B \to bb$$

# Remove repeated productions

$$S \rightarrow aA \mid aB \mid \cancel{aA}$$

$$A \rightarrow a$$

$$B \rightarrow bb$$

Final grammar

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$B \rightarrow bb$$

# Useless Productions

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

$$S \rightarrow A$$

sonuçlanamaz

$$A \rightarrow aA$$ Useless Production

Some derivations never terminate...

$$S \Rightarrow A \Rightarrow aA \Rightarrow aaA \Rightarrow \ldots \Rightarrow aa \ldots aA \Rightarrow \ldots$$

Another grammar:

$$S \to A$$

$$A \to aA$$

$$A \to \lambda$$

$$B \to bA$$

→ B'ye hiç gelmeyeceğiz

Useless Production

Not reachable from S

## In general:

If there is a derivation

$$S \Rightarrow \ldots \Rightarrow xAy \Rightarrow \ldots \Rightarrow w \quad \in L(G)$$

consists of terminals

Then variable $A$ is useful

Otherwise, variable $A$ is useless

A production $A \rightarrow x$ is useless
if any of its variables is useless

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

| Variables | | Productions |
|---|---|---|
| | $\cancel{S \rightarrow A}$ | useless |
| useless | $\cancel{A \rightarrow aA}$ | useless |
| useless | $\cancel{B \rightarrow C}$ | useless |
| useless | $\cancel{C \rightarrow D}$ | useless |

# Removing Useless Variables and Productions

Example Grammar:

$$S \to aS \mid A \mid \cancel{C}$$

$$A \to a$$

$$\cancel{B \to aa}$$

$$\cancel{C \to aCb}$$

**First:** find all variables that can produce strings with only terminals or $\varepsilon$ (possible useful variables)

$$S \to aS \mid A \mid C$$

$$A \to a$$

$$B \to aa$$

$$C \to aCb$$

Round 1: $\{A, B\}$

(the right hand side of production that has only terminals)

Round 2: $\{A, B, S\}$

(the right hand side of a production has terminals and variables of previous round)

This process can be generalized

Then, remove productions that use variables other than $\{A, B, S\}$

$$S \rightarrow aS \mid A \mid \cancel{C}$$
$$A \rightarrow a$$
$$B \rightarrow aa$$
$$\cancel{C \rightarrow aCb}$$

$$S \rightarrow aS \mid A$$
$$A \rightarrow a$$
$$B \rightarrow aa$$

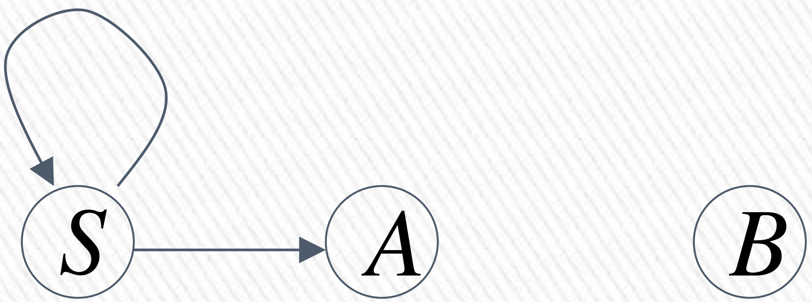
**Second:** Find all variables reachable from $S$

Use a Dependency Graph where nodes are variables

$$S \to aS \mid A$$

$$A \to a$$

$$B \to aa$$




unreachable

Keep only the variables reachable from S

$$S \to aS \mid A$$
$$A \to a$$
$$\cancel{B \to aa}$$

Final Grammar

$$S \to aS \mid A$$
$$A \to a$$

Contains only useful variables

# Removing All

» **Step 1:** Remove Nullable Variables

↳ $A \to \varepsilon$

» **Step 2:** Remove Unit-Productions

↳ $A \to B$

» **Step 3:** Remove Useless Variables

↳ $A \to Aa$

This sequence guarantees that unwanted variables and productions are removed

$S \to ABAC$

$A \to aA \mid \varepsilon$

$B \to bB \mid \varepsilon$

$C \to c$

1 Remove null productions

$A \to \varepsilon,\ B \to \varepsilon$

$S \to ABAC \mid ABC \mid BAC \mid BC \mid C \mid AAC \mid AC$

$A \to aA \mid a$

$B \to bB \mid b$

$\cancel{C \to c}$

$S \to ABAc \mid ABc \mid BAc \mid Bc \mid c \mid AAc \mid Ac$

$A \to aA \mid a$

$B \to bB \mid b$

$S \to ABAC \mid ABC \mid BAC \mid BC \mid AAC \mid AC \mid C$

$A \to aA \mid a$

$B \to aB \mid b$

$\cancel{C \to c}$

# Normal Forms
# for
# Context-free Grammars

# Chomsky Normal Form

Each productions has form:

$$A \to BC \qquad \text{or} \qquad A \to a$$

variable variable terminal

# Examples:

$$S \to AS$$
$$S \to a$$
$$A \to SA$$
$$A \to b$$

$S \to AV_1$

$V_1 \to AS$

$T_1 \to a$

$$S \to AS$$
$$S \to AAS$$
$$A \to SA$$
$$A \to aa$$

$A \to T_1T_1$

Chomsky Normal Form

Not Chomsky Normal Form

# Conversion to Chomsky Normal Form

» Example:

$$S \rightarrow ABa$$

$$A \rightarrow aab$$

$$B \rightarrow Ac$$

**Not** in Chomsky Normal Form

$V_1 \rightarrow a$

$V_2 \rightarrow B V_1$

We will convert it to Chomsky Normal Form

$V_3 \rightarrow b$

$V_4 \rightarrow V_1 V_3$

$V_5 \rightarrow c$

Introduce new variables for the terminals:

$$T_a, T_b, T_c$$

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

$$\Longrightarrow$$

$$S \to ABT_a$$

$$A \to T_a T_a T_b$$

$$B \to AT_c$$

$$\boxed{\begin{array}{l} T_a \to a \\ T_b \to b \\ T_c \to c \end{array}}$$

Introduce new intermediate variable $V_1$ to break first production:

$$\boxed{S \rightarrow ABT_a}$$
$$A \rightarrow T_a T_a T_b$$
$$B \rightarrow AT_c$$
$$T_a \rightarrow a$$
$$T_b \rightarrow b$$
$$T_c \rightarrow c$$

$$\Longrightarrow$$

$$\boxed{\begin{aligned} S &\rightarrow AV_1 \\ V_1 &\rightarrow BT_a \end{aligned}}$$
$$A \rightarrow T_a T_a T_b$$
$$B \rightarrow AT_c$$
$$T_a \rightarrow a$$
$$T_b \rightarrow b$$
$$T_c \rightarrow c$$

Introduce intermediate variable: $V_2$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$\boxed{A \to T_a T_a T_b}$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$\boxed{\begin{array}{l} A \to T_a V_2 \\ V_2 \to T_a T_b \end{array}}$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

# Final grammar in Chomsky Normal Form:

Initial grammar

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

$$S \to AV_1$$

$$V_1 \to BT_a$$

$$A \to T_a V_2$$

$$V_2 \to T_a T_b$$

$$B \to AT_c$$

$$T_a \to a$$

$$T_b \to b$$

$$T_c \to c$$

## In general:

From any context-free grammar
(which doesn't produce $\varepsilon$ )
not in Chomsky Normal Form

we can obtain:

an equivalent grammar
in Chomsky Normal Form

# The Procedure

First remove:

Nullable variables

Unit productions

(Useless variables optional)

Then, for every symbol $a$:

New variable: $T_a$

Add production $T_a \rightarrow a$

---

In productions with length at least 2
replace $a$ with $T_a$

Productions of form $A \rightarrow a$
do not need to change!

Replace any production $A \to C_1 C_2 \cdots C_n$

with

$$A \to C_1 V_1$$

$$V_1 \to C_2 V_2$$

$$\cdots$$

$$V_{n-2} \to C_{n-1} C_n$$

New intermediate variables: $V_1, V_2, \ldots, V_{n-2}$

# Observations

- Chomsky normal forms are good for parsing and proving theorems
- It is easy to find the Chomsky normal form for any context-free grammar

# Greinbach Normal Form

All productions have form:

$$A \to a\, V_1 V_2 \cdots V_k \qquad k \geq 0$$

symbol

variables

# Examples:

$$S \to cAB$$

$$A \to aA \mid bB \mid b$$

$$B \to b$$

Greinbach Normal Form

$b$ ✓

$T_1 = b$

$S \to aT_1ST_1$

$$S \to abSb$$

$$S \to aa$$

$T_2 = a$

$S \to aT_2$

Not Greinbach Normal Form

# Conversion to Greinbach Normal Form:

$V_1$ $V_1$

$$S \to abSb$$

$V_2$

$$S \to a\cancel{a}$$

$V_1 \Rightarrow b$

$V_2 \to a$

$S$

$$S \to aT_bST_b$$

$$S \to aT_a$$

$$T_a \to a$$

$$T_b \to b$$

Greinbach Normal Form

# Observations

- Greinbach normal forms are very good for parsing strings (better than Chomsky Normal Forms)

- However, it is difficult to find the Greinbach normal of a grammar

# BLM2502 Theory of Computation

?