Part X.
Normal Forms and Properties of CFLs

Chomsky Normal Form (CNF)

Definition: Let $G = (N, T, P, S)$ be a CFG.
G is in *Chomsky normal form* if every rule in P has one of these forms

- $\boldsymbol{A \to BC}$, where $A, B, C \in N$;
- $\boldsymbol{A \to a}$, where $A \in N$, $a \in T$;

Example:

$G = (N, T, P, S)$, where $N = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{S}\}$, $T = \{\boldsymbol{a}, \boldsymbol{b}\}$,
$P = \{\boldsymbol{S \to CB}, \boldsymbol{C \to AS}, \boldsymbol{S \to AB}, \boldsymbol{A \to a}, \boldsymbol{B \to b}\}$
is in Chomsky normal form.

Note: $L(G) = \{a^n b^n\text{: } n \geq 1\}$

Greibach Normal Form (GNF)

Definition: Let $G = (N, T, P, S)$ be a CFG. G is in *Greibach normal form* if every rule in P is of this form

- $\boldsymbol{A \to ax}$, where $A \in N$, $a \in T$, $x \in N^*$

Example:

$G = (N, T, P, S)$, where $N = \{\boldsymbol{B}, \boldsymbol{S}\}$, $T = \{\boldsymbol{a}, \boldsymbol{b}\}$,
$P = \{\boldsymbol{S \to aSB}, \boldsymbol{S \to aB}, \boldsymbol{B \to b}\}$
is in Greibach normal form.

Note: $L(G) = \{a^n b^n: n \geq 1\}$

Generative Power of Normal Forms

Theorem: For every CFG G, there is an equivalent grammar G' in Chomsky normal form.

Proof: See page 348 in [Meduna: Automata and Languages]

Theorem: For every CFG G, there is an equivalent grammar G' in Greibach normal form.

Proof: See page 376 in [Meduna: Automata and Languages]

Note: Main properties of CNF and GNF:

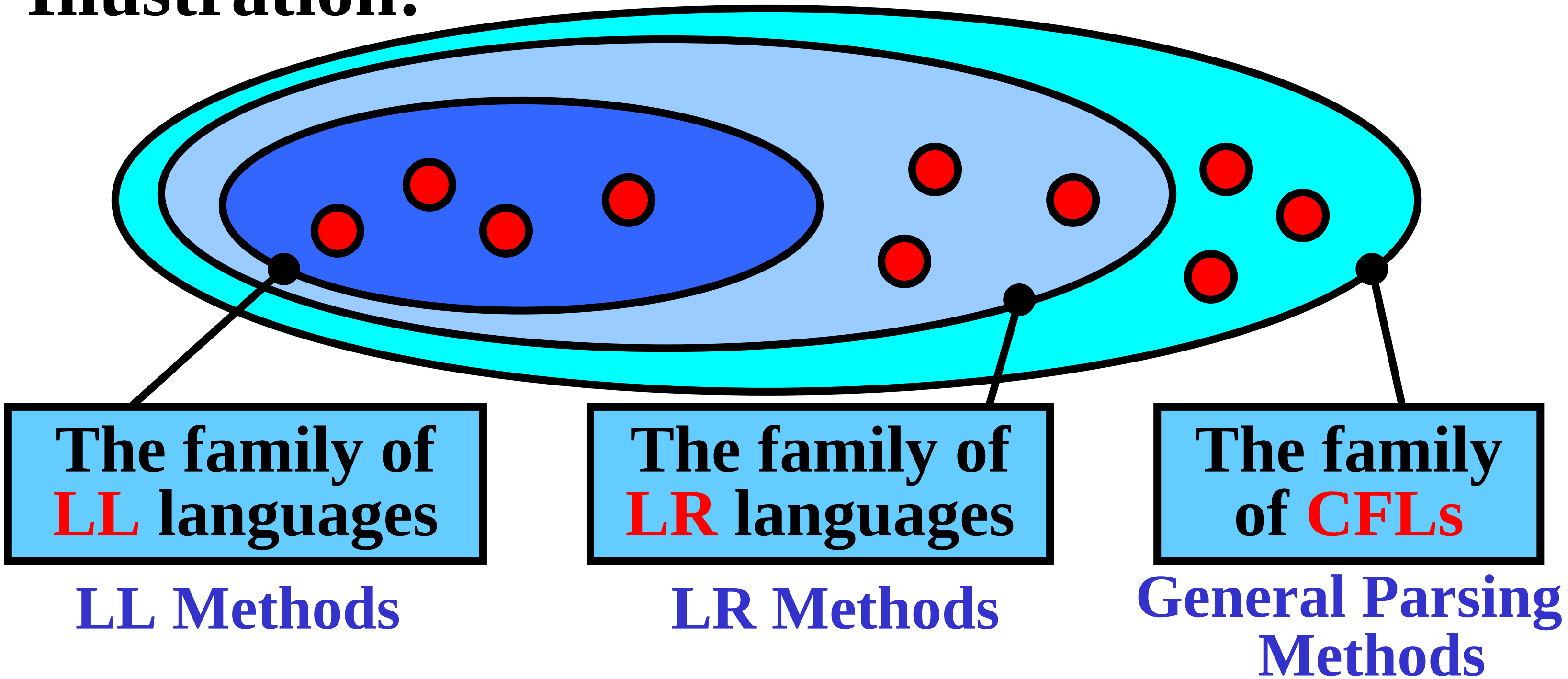
CNF: **if** $S \Rightarrow^n w$; $w \in T^*$ **then** $n = 2|w| - 1$

GNF: **if** $S \Rightarrow^n w$; $w \in T^*$ **then** $n = |w|$

General Parsing Methods

- ***G**eneral **P**arsing methods* (GP) are applicable to all context-free languages (CFLs)

Illustration:

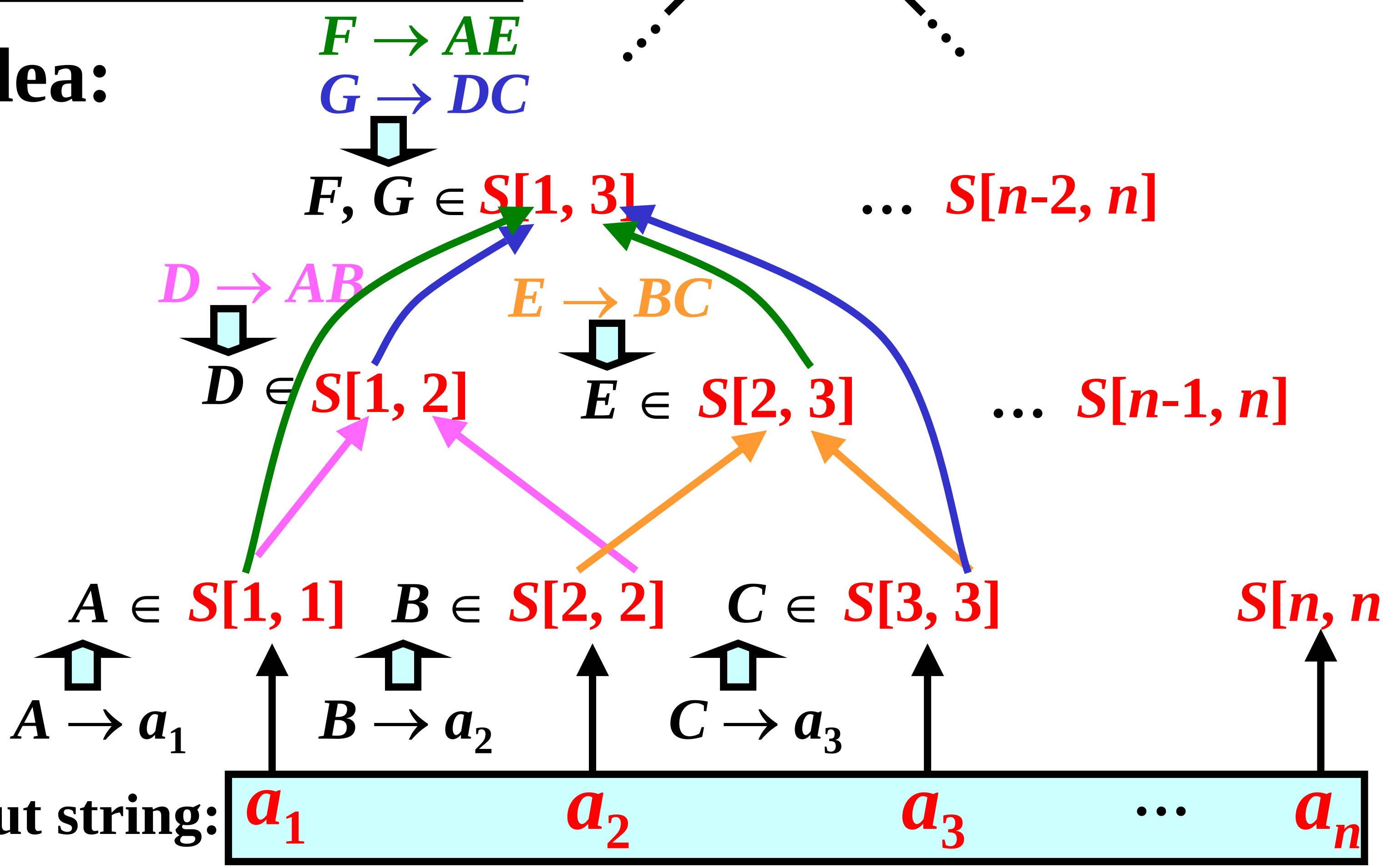


- **Note:** The family of **LR** languages = the family of a **deterministic CFL**

GP Based on Chomsky Normal Form

if $S \in S[1, n]$ then
$S \Rightarrow^* a_1 \ldots a_n$

Idea:



• **Input string:**

Algorithm: GP Based on CNF

- **Input:** $G = (N, T, P, S)$ in CNF, $\boldsymbol{w} = \boldsymbol{a_1 \ldots a_n}$
- **Output:** **YES** if $w \in L(G)$
  NO if $w \notin L(G)$

- **Method:**
- **for each** $\boldsymbol{a_i}$, $\boldsymbol{i} = 1, \ldots, \boldsymbol{n}$ **do**
  $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{i}] := \{\boldsymbol{A} : \boldsymbol{A} \rightarrow \boldsymbol{a_i} \in \boldsymbol{P}\}$
- **Apply the following rule until no $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{k}]$ can be changed:**
  if $\boldsymbol{A} \rightarrow \boldsymbol{BC} \in \boldsymbol{P}$, $\boldsymbol{B} \in \boldsymbol{S}\,[\boldsymbol{i}, \boldsymbol{j}]$, $\boldsymbol{C} \in \boldsymbol{S}\,[\boldsymbol{j}+1, \boldsymbol{k}]$,
  where $1 \leq \boldsymbol{i} \leq \boldsymbol{j} < \boldsymbol{k} \leq \boldsymbol{n}$ **then** add $\boldsymbol{A}$ to $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{k}]$
- **if** $\boldsymbol{S} \in \boldsymbol{S}[1, \boldsymbol{n}]$ **then** write ('**YES**')
  else write ('**NO**')

GP Based on CNF: Example 1/5

$G = (N, T, P, S)$, where $N = \{A, B, C, S\}$, $T = \{a, b\}$,
$P = \{S \to AC, C \to SB, A \to a, B \to b, S \to c\}$
Question: $aacbb \in L(G)$**?**

$S[1, 1]=\{A\}$ $S[2, 2]=\{A\}$ $S[3, 3]=\{S\}$ $S[4, 4]=\{B\}$ $S[5, 5]=\{B\}$

$A \to a$ $A \to a$ $S \to c$ $B \to b$ $B \to b$

a a c b b

GP Based on CNF: Example 2/5

$G = (N, T, P, S)$, where $N = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{S}\}$, $T = \{\mathbf{a}, \mathbf{b}\}$,
$P = \{\mathbf{S} \rightarrow \mathbf{AC}, \mathbf{C} \rightarrow \mathbf{SB}, \mathbf{A} \rightarrow \mathbf{a}, \mathbf{B} \rightarrow \mathbf{b}, \mathbf{S} \rightarrow \mathbf{c}\}$
Question: $\mathbf{aacbb} \in L(G)$**?**

$? \cancel{\rightarrow} AA$ $\quad$ $? \cancel{\rightarrow} AS$ $\quad$ $C \rightarrow SB$ $\quad$ $? \cancel{\rightarrow} BB$

$S[1, 2]=\emptyset$ $\quad$ $S[2, 3]=\emptyset$ $\quad$ $S[3, 4]=\{C\}$ $\quad$ $S[4, 5]=\emptyset$

$S[1, 1]=\{A\}$ $\quad$ $S[2, 2]=\{A\}$ $\quad$ $S[3, 3]=\{S\}$ $\quad$ $S[4, 4]=\{B\}$ $\quad$ $S[5, 5]=\{B\}$

a	***a***	***c***	***b***	***b***

GP Based on CNF: Example 3/5

$G = (N, T, P, S)$, where $N = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{S}\}$, $T = \{\boldsymbol{a}, \boldsymbol{b}\}$,
$P = \{\boldsymbol{S} \rightarrow \boldsymbol{AC}, \boldsymbol{C} \rightarrow \boldsymbol{SB}, \boldsymbol{A} \rightarrow \boldsymbol{a}, \boldsymbol{B} \rightarrow \boldsymbol{b}, \boldsymbol{S} \rightarrow \boldsymbol{c}\}$
Question: $\boldsymbol{aacbb} \in L(G)$?

$\boldsymbol{S} \rightarrow \boldsymbol{AC}$

$\boldsymbol{S}[1, 3] = \emptyset$ $\boldsymbol{S}[2, 4] = \{\boldsymbol{S}\}$ $\boldsymbol{S}[3, 5] = \emptyset$

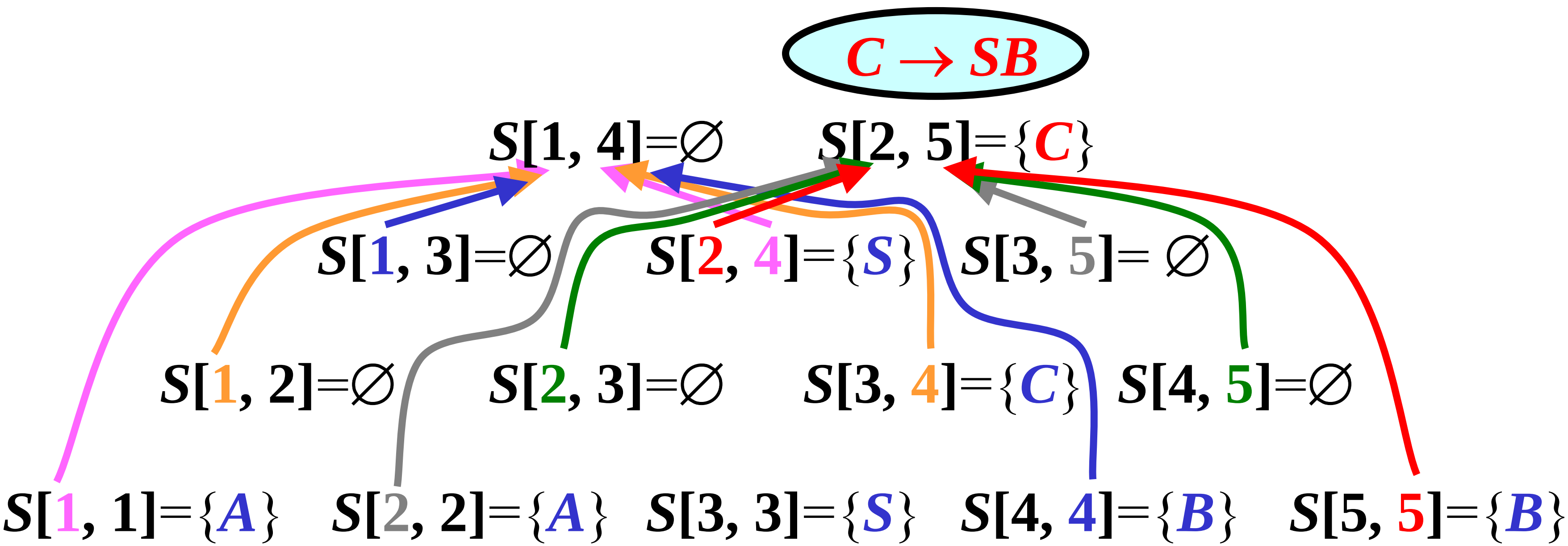
$\boldsymbol{S}[1, 2] = \emptyset$ $\boldsymbol{S}[2, 3] = \emptyset$ $\boldsymbol{S}[3, 4] = \{\boldsymbol{C}\}$ $\boldsymbol{S}[4, 5] = \emptyset$

$\boldsymbol{S}[1, 1] = \{\boldsymbol{A}\}$ $\boldsymbol{S}[2, 2] = \{\boldsymbol{A}\}$ $\boldsymbol{S}[3, 3] = \{\boldsymbol{S}\}$ $\boldsymbol{S}[4, 4] = \{\boldsymbol{B}\}$ $\boldsymbol{S}[5, 5] = \{\boldsymbol{B}\}$

a	***a***	***c***	***b***	***b***

GP Based on CNF: Example 4/5

$G = (N, T, P, S)$, where $N = \{A, B, C, S\}$, $T = \{a, b\}$,
$P = \{S \rightarrow AC, C \rightarrow SB, A \rightarrow a, B \rightarrow b, S \rightarrow c\}$
Question: $aacbb \in L(G)$?



a	a	c	b	b

GP Based on CNF: Example 5/5

$G = (N, T, P, S)$, where $N = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{S}\}$, $T = \{\boldsymbol{a}, \boldsymbol{b}\}$,
$P = \{\boldsymbol{S} \rightarrow \boldsymbol{AC}, \boldsymbol{C} \rightarrow \boldsymbol{SB}, \boldsymbol{A} \rightarrow \boldsymbol{a}, \boldsymbol{B} \rightarrow \boldsymbol{b}, \boldsymbol{S} \rightarrow \boldsymbol{c}\}$
Question: $\boldsymbol{aacbb} \in L(G)$**?**

$S \rightarrow AC$

$\boldsymbol{S}[1, 5]=\{S\}$

$S \in \boldsymbol{S}[1, 5]$ **YES**

$\boldsymbol{S}[1, 4]=\varnothing$ $\boldsymbol{S}[2, 5]=\{C\}$

$\boldsymbol{S}[1, 3]=\varnothing$ $\boldsymbol{S}[2, 4]=\{S\}$ $\boldsymbol{S}[3, 5]=\varnothing$

$\boldsymbol{S}[1, 2]=\varnothing$ $\boldsymbol{S}[2, 3]=\varnothing$ $\boldsymbol{S}[3, 4]=\{C\}$ $\boldsymbol{S}[4, 5]=\varnothing$

$\boldsymbol{S}[1, 1]=\{A\}$ $\boldsymbol{S}[2, 2]=\{A\}$ $\boldsymbol{S}[3, 3]=\{S\}$ $\boldsymbol{S}[4, 4]=\{B\}$ $\boldsymbol{S}[5, 5]=\{B\}$

a	***a***	***c***	***b***	***b***

Pumping Lemma for CFL

- Let L be CFL. Then, there exists $k \geq 1$ such that:

if $z \in L$ and $|z| \geq k$ **then** there exist u, v, w, x, y so $z = uvwxy$ and

1) $vx \neq \varepsilon$ **2)** $|vwx| \leq k$ **3)** for each $m \geq 0$, $uv^m wx^m y \in L$

Example:

$G = (\{S, A\}, \{a, b, c\}, \{S \to aAa, A \to bAb, A \to c\}, S)$

generate $L(G) = \{ab^n cb^n a : n \geq 0\}$, so $L(G)$ is CFL.

There is $k = 5$ such that **1)**, **2)** and **3)** holds:

- for $z = abcba$: $z \in L(G)$ and $|z| \geq 5$:

$u\,v\,w\,x\,y$

$vx = bb \neq \varepsilon$

$|vwx| = 3$: $1 \leq 3 \leq 5$

$uv^0wx^0y = ab^0cb^0a = aca \in L(G)$

$uv^1wx^1y = ab^1cb^1a = abcba \in L(G)$

$uv^2wx^2y = ab^2cb^2a = abbcbba \in L(G)$

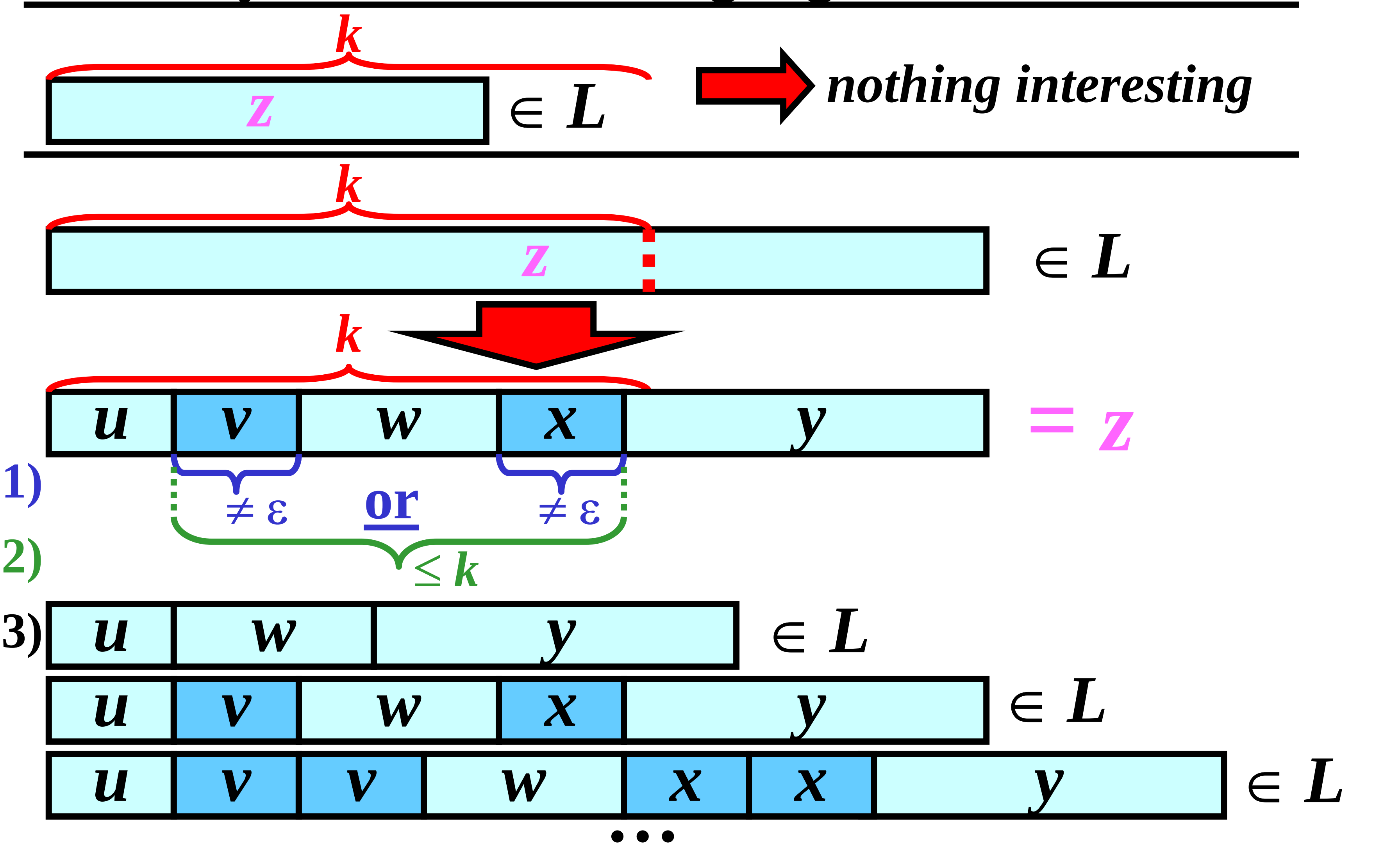
⋮

- for $z = abbcbba$: $z \in L(G)$ and $|z| \geq 5$:

⋮

Pumping Lemma: Illustration

- L = any context-free language:



Pumping Lemma: Application

- Based on the pumping lemma for CFL, we often make a proof by contradiction to demonstrate that a language is **not** a CFL.

Assume that L is a CFL.

Consider the PL constant $\boldsymbol{k}$ and select $\boldsymbol{z} \in L$, whose length depends on $\boldsymbol{k}$ so $|\boldsymbol{z}| \geq \boldsymbol{k}$ is surely true.

For <u>all</u> decompositions of $\boldsymbol{z}$ into $\boldsymbol{uvwxy}$: $vx \neq \varepsilon$, $|vwx| \leq k$, show that

$$\left.\begin{array}{l}\text{there exists } m \geq 0 \text{ such that } \boldsymbol{uv^m wx^m y} \notin \boldsymbol{L};\\ \text{from the pumping lemma, } \boldsymbol{uv^m wx^m y} \in \boldsymbol{L}\end{array}\right\} \textbf{contradiction}$$

false assumption

Therefore,
L is not a CFL

Pumping Lemma: Example 1/2

Prove that $L = \{a^n b^n c^n : n \geq 1\}$ is not CFL.

1) Assume that L is a CFL. Let $\boldsymbol{k} \geq 1$ be the pumping lemma constant for L.

2) Let $\boldsymbol{z} = \boldsymbol{a^k b^k c^k}$: $\boldsymbol{a^k b^k c^k} \in L$, $|\boldsymbol{z}| = |\boldsymbol{a^k b^k c^k}| = 3\boldsymbol{k} \geq \boldsymbol{k}$

3) All decompositions of $\boldsymbol{z}$ into $\boldsymbol{uvwxy}$; $vx \neq \varepsilon$, $|vwx| \leq k$:

$$\overbrace{aaaaa\ldots aa}^{k}\overbrace{bb\ldots bb\ldots bb}^{k}\overbrace{cc\ldots ccccc}^{k}$$

a) $vwx \in \{a\}^*\{b\}^*$, $vx \neq \varepsilon$

b) $vwx \in \{b\}^*\{c\}^*$, $vx \neq \varepsilon$

Pumping Lemma: Example 2/2

a) $vwx \in \{a\}^*\{b\}^*$:

- Pumping lemma: $uv^0wx^0y \in L$

$$\overbrace{\underbrace{a}_{u}\underbrace{a\ldots aabb\ldots b}_{vwx}\underbrace{bcc\ldots cc}_{y}}^{k\qquad\qquad k\qquad\qquad k}$$

- $uv^0wx^0y = uwy = \underbrace{a}_{u}\,a\,\underbrace{\ldots aabb\ldots}_{w}\,b\,\underbrace{bcc\ldots cc}_{y} \notin L$

Note: uwy contains k cs, but fewer than k as or bs.

---

b) $vwx \in \{b\}^*\{c\}^*$:

- Pumping lemma: $uv^0wx^0y \in L$

$$\overbrace{\underbrace{aa\ldots aab}_{u}\underbrace{b\ldots bbcc\ldots c}_{vwx}\underbrace{c}_{y}}^{k\qquad\qquad k\qquad\qquad k}$$

- $uv^0wx^0y = uwy = \underbrace{aa\ldots aab}_{u}\,b\,\underbrace{\ldots bbcc\ldots}_{w}\,c\,\underbrace{c}_{y} \notin L$

Note: uwy contains k as, but fewer than k bs or cs.

All these decompositions lead to a contradiction!

---

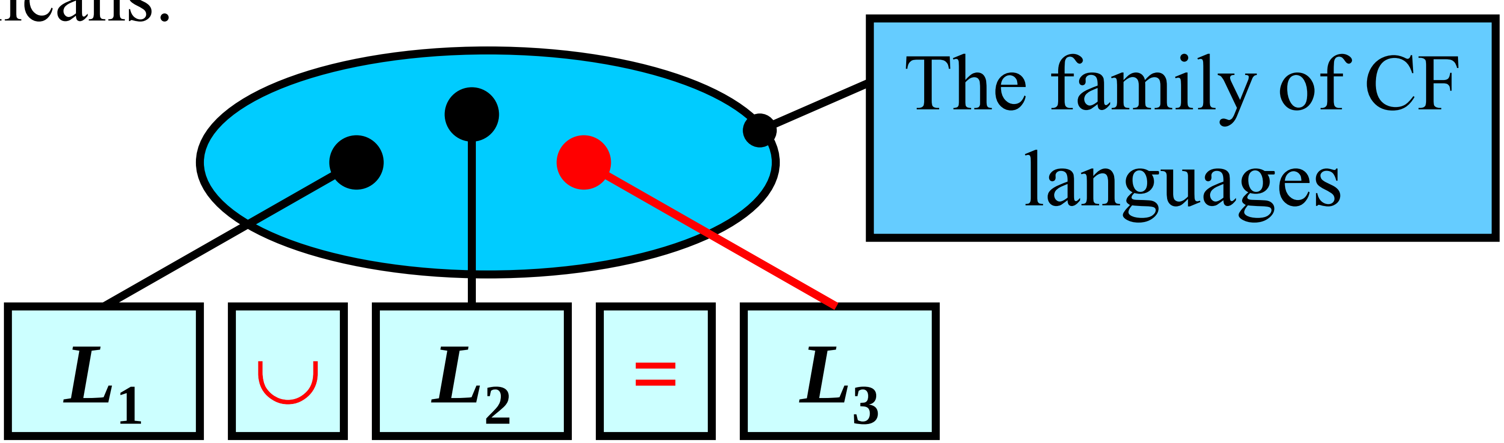
4) Therefore, L is not a CFL.

Closure properties of CFL

Definition: The family of CFLs is closed under an operation ***o*** if the language resulting from the application of ***o*** to **any** CFLs is a CFL as well.

Illustration:

- The family of CF languages is closed under ***union***. It means:



Algorithm: CFG for Union

- **Input:** Grammars $G_1 = (N_1, T, P_1, S_1)$ and
  $G_2 = (N_2, T, P_2, S_2)$;
- **Output:** Grammar $G_u = (N, T, P, S)$ such that
  $L(G_u) = L(G_1) \cup L(G_2)$

- **Method:**
- let $S \notin N_1 \cup N_2$, let $N_1 \cap N_2 = \emptyset$:
  - $N := \{S\} \cup N_1 \cup N_2$;
  - $P := \{S \to S_1, S \to S_2\} \cup P_1 \cup P_2$;

Algorithm: CFG for Concatenation

- **Input:** $G_1 = (N_1, T, P_1, S_1)$ and
  $G_2 = (N_2, T, P_2, S_2)$;
- **Output:** $G_c = (N, T, P, S)$ such that
  $L(G_c) = L(G_1) \,.\, L(G_2)$

- **Method:**
- let $S \notin N_1 \cup N_2$, let $N_1 \cap N_2 = \varnothing$:
  - $N := \{S\} \cup N_1 \cup N_2$;
  - $P := \{S \to S_1S_2\} \cup P_1 \cup P_2$;

Algorithm: CFG for Iteration

- **Input:** $G = (N_1, T, P_1, S_1)$
- **Output:** $G_i = (N, T, P, S)$ such that $L(G_i) = L(G)^*$

- **Method:**
- let $S \notin N_1$:
  - $N := \{S\} \cup N_1$;
  - $P := \{S \rightarrow S_1S, S \rightarrow \varepsilon\} \cup P_1$;

Closure properties

Theorem: The family of CFLs is closed under **union**, **concatenation**, **iteration**.

Proof:

- Let L_1, L_2 be two **CFLs.**
- Then, there exist two CFGs G_1, G_2 such that $L(G_1) = \boldsymbol{L_1}$, $L(G_2) = \boldsymbol{L_2}$;
- Construct grammars
  - G_u such that $L(G_u) = L(G_1) \cup L(G_2)$
  - G_c such that $L(G_c) = L(G_2)\ .\ L(G_2)$
  - G_i such that $L(G_i) \ = L(G_1)^*$

  by using the previous three algorithms
- Every CFG denotes CFL, so
- $L_1 L_2$, $L_1 \cup L_2$, L_1^* are **CFLs.**

Intersection: Not Closed

Theorem: The family of CFLs is **not** closed under **intersection**.

Proof:

- The intersection of some CFLs is not a CFL:

- $L_1 = \{a^m b^n c^n : m, n \geq 1\}$ is a CFL
- $L_2 = \{a^n b^n c^m : m, n \geq 1\}$ is a CFL
- $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 1\}$ is not a CFL
  (proof based on the pumping lemma) *QED*

Complement: Not Closed

Theorem: The family of CFLs is not closed under **complement**.

Proof by contradiction:

- Assume that family of CFLs is closed under complement.
- $L_1 = \{a^m b^n c^n : m, n \geq 1\}$ is a **CFL**
- $L_2 = \{a^n b^n c^m : m, n \geq 1\}$ is a **CFL**
- $\overline{L_1}, \overline{L_2}$ are **CFLs**
- $\overline{L_1} \cup \overline{L_2}$ is a **CFL** (the family of CFLs is closed under union)
- $\overline{\overline{L_1} \cup \overline{L_2}}$ is a **CFL** (assumption)
- DeMorgan's law implies $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 1\}$ is a **CFL**
- $\{a^n b^n c^n : n \geq 1\}$ is not a **CFL** $\Rightarrow$ **Contradiction**

Main Decidable Problems

1. Membership problem:

- **Instance:** CFG G, $w \in \Sigma^*$;**Question:** $w \in L(G)$**?**

2. Emptiness problem:

- **Instance:** CFG G; **Question:** $L(G) = \varnothing$**?**

3. Finiteness problem:

- **Instance:** CFG G; **Question:** Is $L(G)$ finite**?**

Algorithm: Membership

- **Input:** CFG $G = (N, T, P, S)$ in Chomsky normal form; $w \in T^+$
- **Output:** **YES** if $w \in L(G)$
  NO if $w \notin L(G)$

- **Method I:**
- **if** $S \Rightarrow^{n} w$, where $1 \leq n \leq 2|w| - 1$, **then** write ('**YES**')
  else write ('**NO**')

- **Method II:**
- See: **The general parsing method based on CNF**

Summary:

The membership problem for CFLs is decidable

Accessible Symbols

Gist: Symbol X is *accessible* if $S \Rightarrow^* \ldots X \ldots$, where S is the start nonterminal.

Definition: Let $G = (N, T, P, S)$ be a CFG. A symbol $X \in N \cup T$ is *accessible* if there exist $u, v \in \Sigma^*$ such that $S \Rightarrow^* uXv$; otherwise, X is *inaccessible*.

Note: Each inaccessible symbol can be removed from CFG

Example:

$G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow SB, S \rightarrow a, A \rightarrow ab, B \rightarrow aB\}, S)$

S - accessible: for $u = \varepsilon$, $v = \varepsilon$: $S \Rightarrow^0 S$

A - **inaccessible**: there is no $u, v \in \Sigma^*$ such that $S \Rightarrow^* uAv$

B - accessible: for $u = S$, $v = \varepsilon$: $S \Rightarrow^1 SB$

a - accessible: for $u = \varepsilon$, $v = \varepsilon$: $S \Rightarrow^1 a$

b - **inaccessible**: there is no $u, v \in \Sigma^*$ such that $S \Rightarrow^* ubv$

Terminating Symbols

Gist: Symbol *X* is *terminating* if *X* derives a terminal string.

Definition: Let $G = (N, T, P, S)$ be a CFG. A symbol $X \in N \cup T$ is *terminating* if there exists $w \in T^*$ such that $X \Rightarrow^* w$; otherwise, X is *nonterminating*

Note: Each nonterminating symbol can be removed from any CFG.

Example:

$G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow SB, S \rightarrow a, A \rightarrow ab, B \rightarrow aB\}, S)$

Symbol S - terminating: for $w = a$: $S \Rightarrow^1 a$

Symbol A - terminating: for $w = ab$: $A \Rightarrow^1 ab$

Symbol B - **nonterminating**: there is no $w \in T^*$ such that $B \Rightarrow^* w$

Symbol a - terminating: for $w = a$: $a \Rightarrow^0 a$

Symbol b - terminating: for $w = b$: $b \Rightarrow^0 b$

Algorithm: Emptiness

- **Input:** CFG $G = (N, T, P, S)$;
- **Output:** **YES** if $L(G) = \emptyset$
  NO if $L(G) \neq \emptyset$

- **Method:**
- **if** S is nonterminating **then** write ('**YES**')
  else write ('**NO**')

Summary:

The emptiness problem for CFLs is decidable

Algorithm: Finiteness

- **Input:** CFG $G = (N, T, P, S)$;
- **Output:** **YES** if $L(G)$ is finite
  NO if $L(G)$ is infinite

- **Method:**
- Let $k = 2^{\text{card}(N)}$
- **if** there exist $z \in L(M)$, $k \le |z| < 2k$ **then** write ('**NO**')
  else write ('**YES**')

Summary:

The finiteness problem for CFLs is decidable

Main Undecidable Problems

1. Equivalence problem:

- **Instance:** CFGs G_1, G_2;**Question:** $L(G_1) = L(G_2)$**?**

2. Ambiguity problem:

- **Instance:** G; **Question:** Is G ambiguous**?**

Note:

It is mathematically proved that there exists no algorithm, which solve these problems in finite time.