Syntax Analysis:

Methods and Theory

Sections 3.2 and 3.3

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

- $A \rightarrow BC$, where $A, B, C \in N$;
- $A \rightarrow a$, where $A \in N$, $a \in T$;

Example:

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

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

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

- $A \rightarrow ax$, where $A \in N$, $a \in T$, $x \in N^*$

Example:

$G = (N, T, P, S)$, where $N = \{B, S\}$, $T = \{a, b\}$, $P = \{S \rightarrow aSB, S \rightarrow aB, B \rightarrow b\}$ 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: Omitted.

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

Proof: Omitted.

Note: Main properties of CNF and GNF:

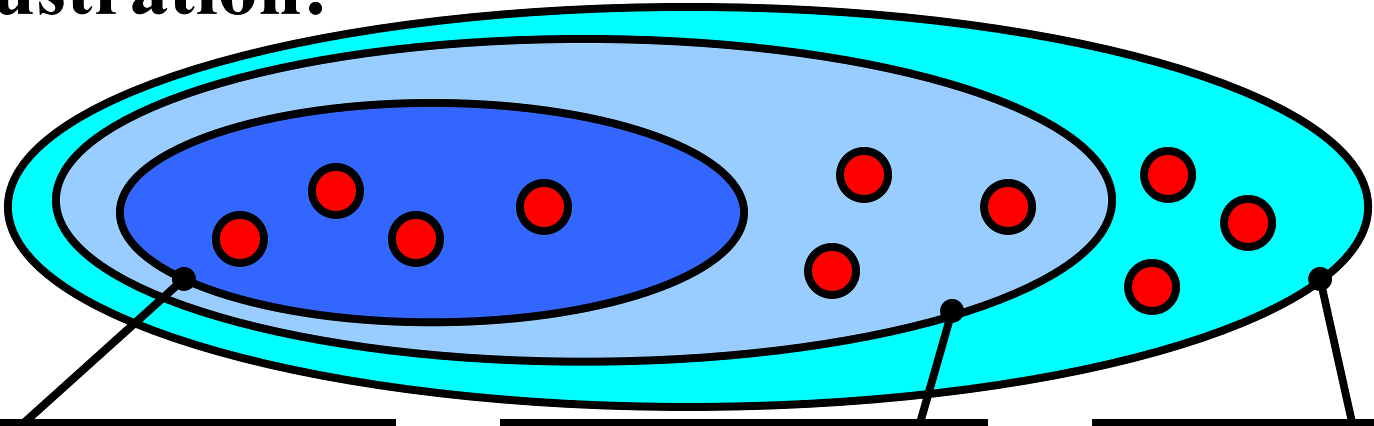
CNF: **if** $S \Rightarrow^{n} w$; $w \in T^*$ **then** $n = 2|w| - 1$

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:



The family of LL languages

LL Methods

The family of LR languages

LR Methods

The family of CFLs

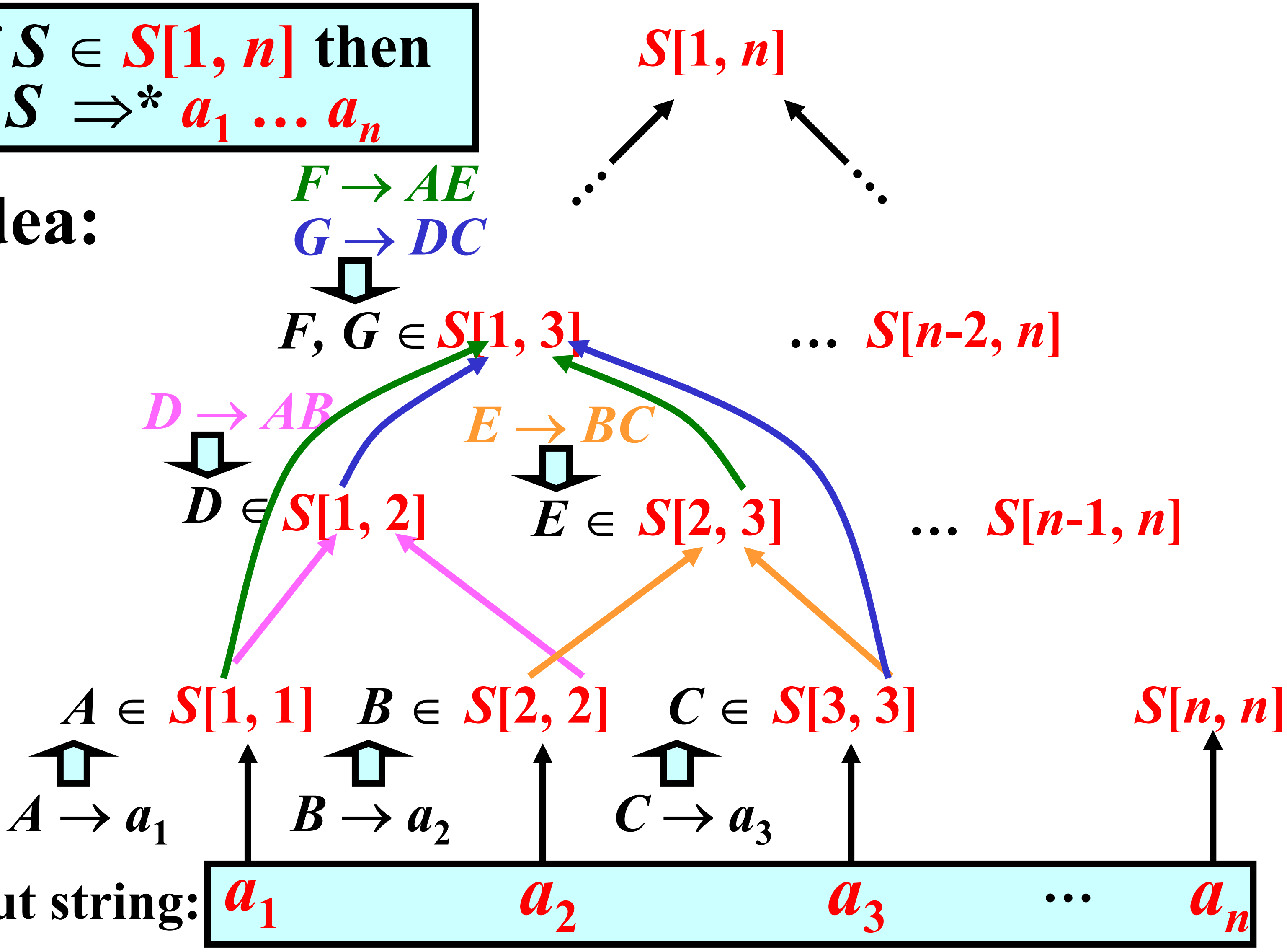
General Parsing Methods

- **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:



$S[1, n]$

$F \rightarrow AE$

$G \rightarrow DC$

$F, G \in S[1, 3]$ … $S[n-2, n]$

$D \rightarrow AB$

$E \rightarrow BC$

$D \in S[1, 2]$ $E \in S[2, 3]$ … $S[n-1, n]$

$A \in S[1, 1]$ $B \in S[2, 2]$ $C \in S[3, 3]$ $S[n, n]$

$A \rightarrow a_1$ $B \rightarrow a_2$ $C \rightarrow a_3$

- **Input string:** a_1 a_2 a_3 … a_n

Algorithm: GP Based on CNF

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

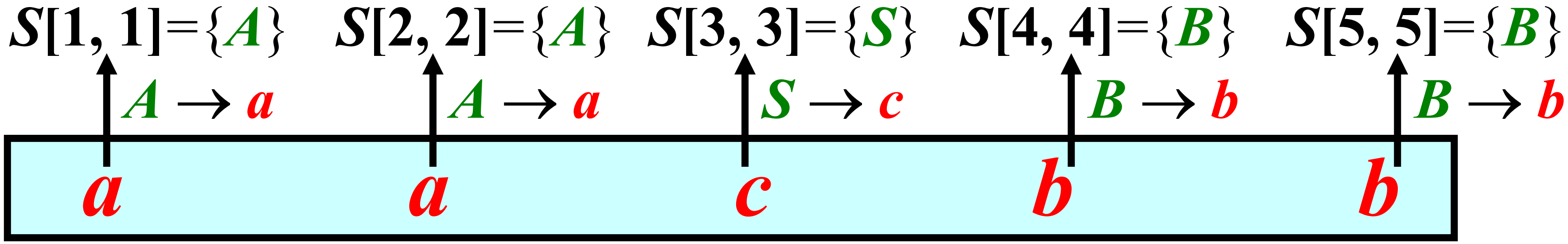
- **Method:**
- **for each** a_i, $i = 1, \ldots, n$ **do**
  $S[i, i] := \{A : A \rightarrow a_i \in P\}$
- **Apply the following rule until no** $S[i, k]$ **can be changed:**
  if $A \rightarrow BC \in P$, $B \in S[i, j]$, $C \in S[j+1, k]$,
  where $1 \leq i \leq j < k \leq n$ **then** add A to $S[i,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

GP Based on CNF: Example 1/5

$G = (N, T, P, S)$, where $N = \{A, B, C, S\}$, $T = \{a, b\}$,

$P = \{S \rightarrow AC, C \rightarrow SB, A \rightarrow a, B \rightarrow b, S \rightarrow c\}$

Question: $aacbb \in L(G)$?



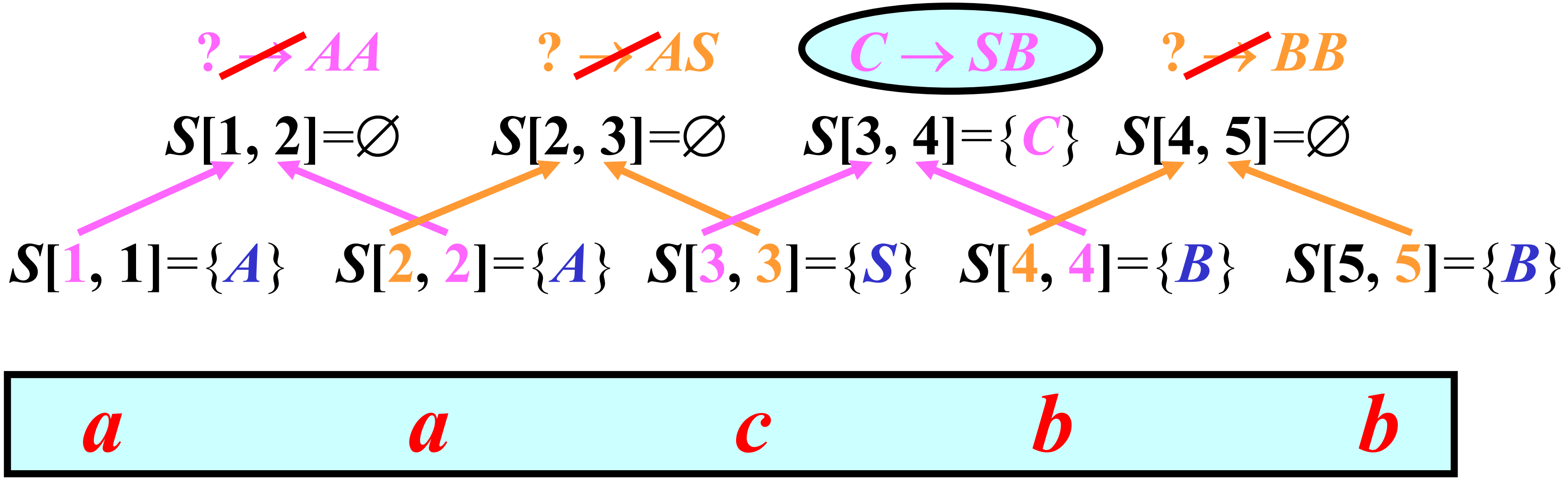
$S[1, 1]=\{A\}$ $S[2, 2]=\{A\}$ $S[3, 3]=\{S\}$ $S[4, 4]=\{B\}$ $S[5, 5]=\{B\}$

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

a a c b b

GP Based on CNF: Example 2/5

$G = (N, T, P, S)$, 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: $aacbb \in L(G)$?



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

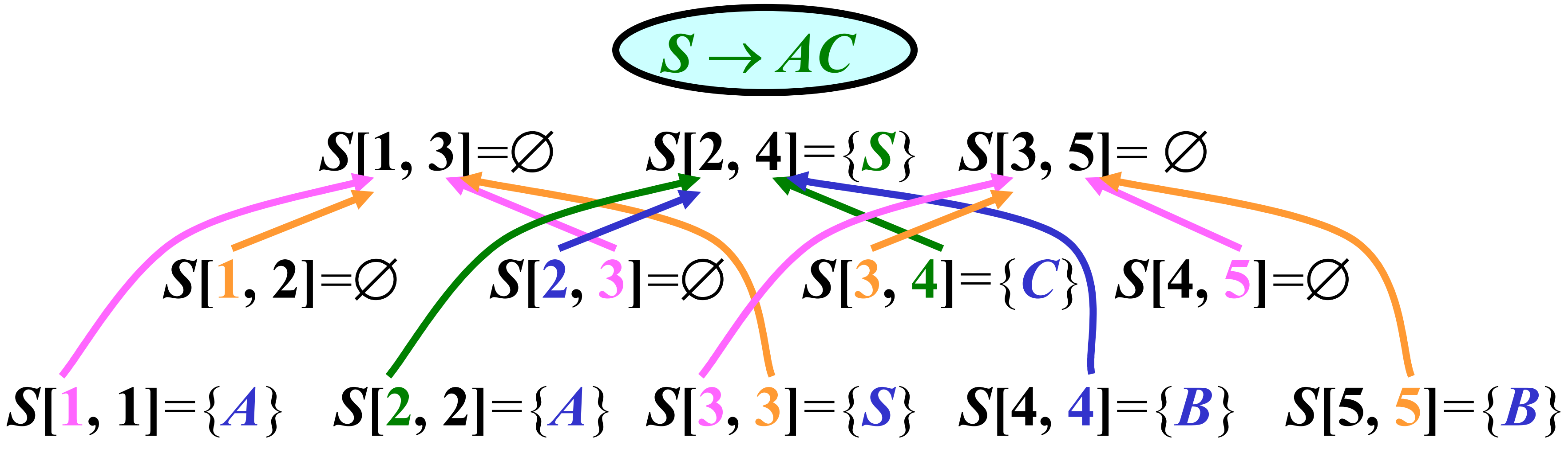
$S[1, 2]=\emptyset$ $S[2, 3]=\emptyset$ $S[3, 4]=\{C\}$ $S[4, 5]=\emptyset$

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

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

GP Based on CNF: Example 3/5

$G = (N, T, P, S)$, 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: $aacbb \in L(G)$?



$S \rightarrow AC$

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

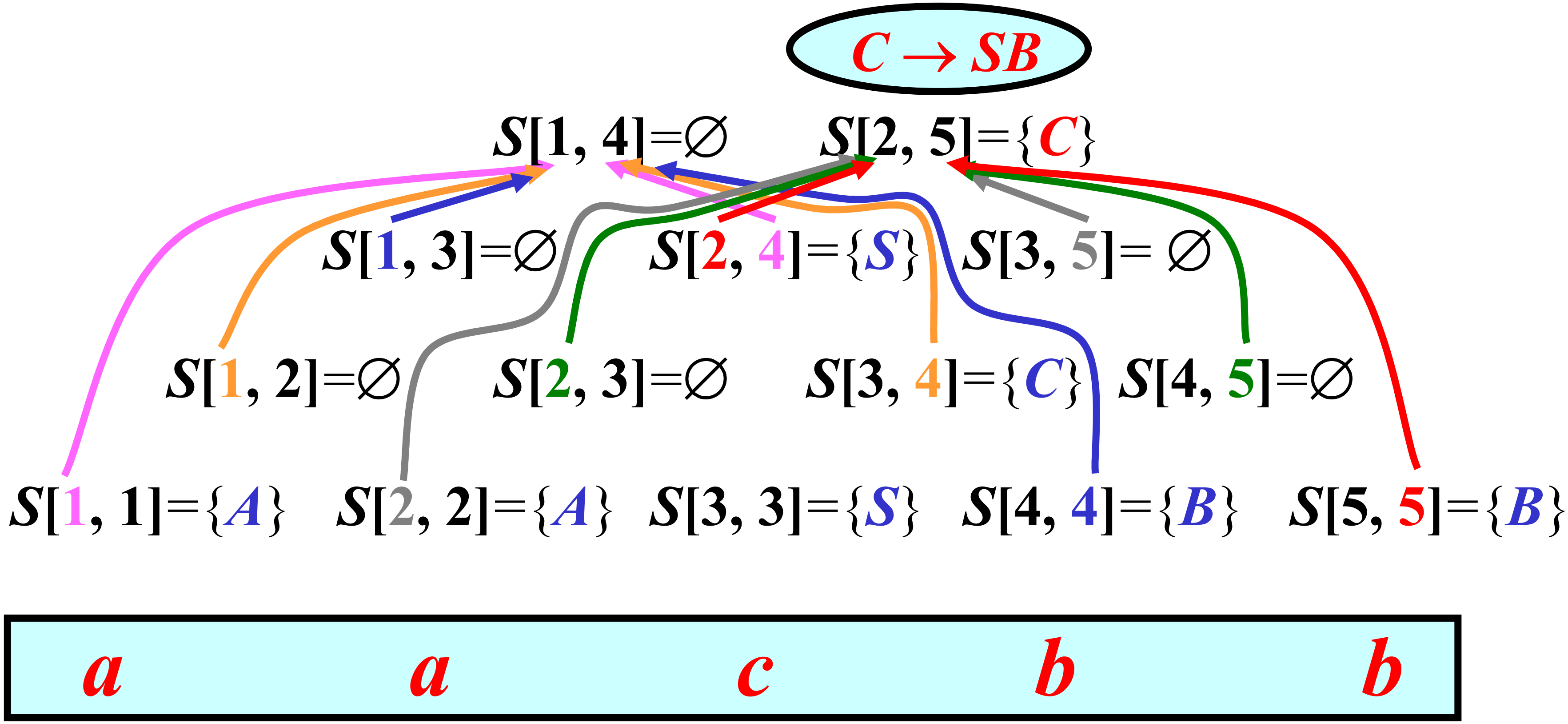
$S[1, 2]=\emptyset$ $S[2, 3]=\emptyset$ $S[3, 4]=\{C\}$ $S[4, 5]=\emptyset$

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

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: $aacbb \in L(G)$?



$C \rightarrow SB$

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

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

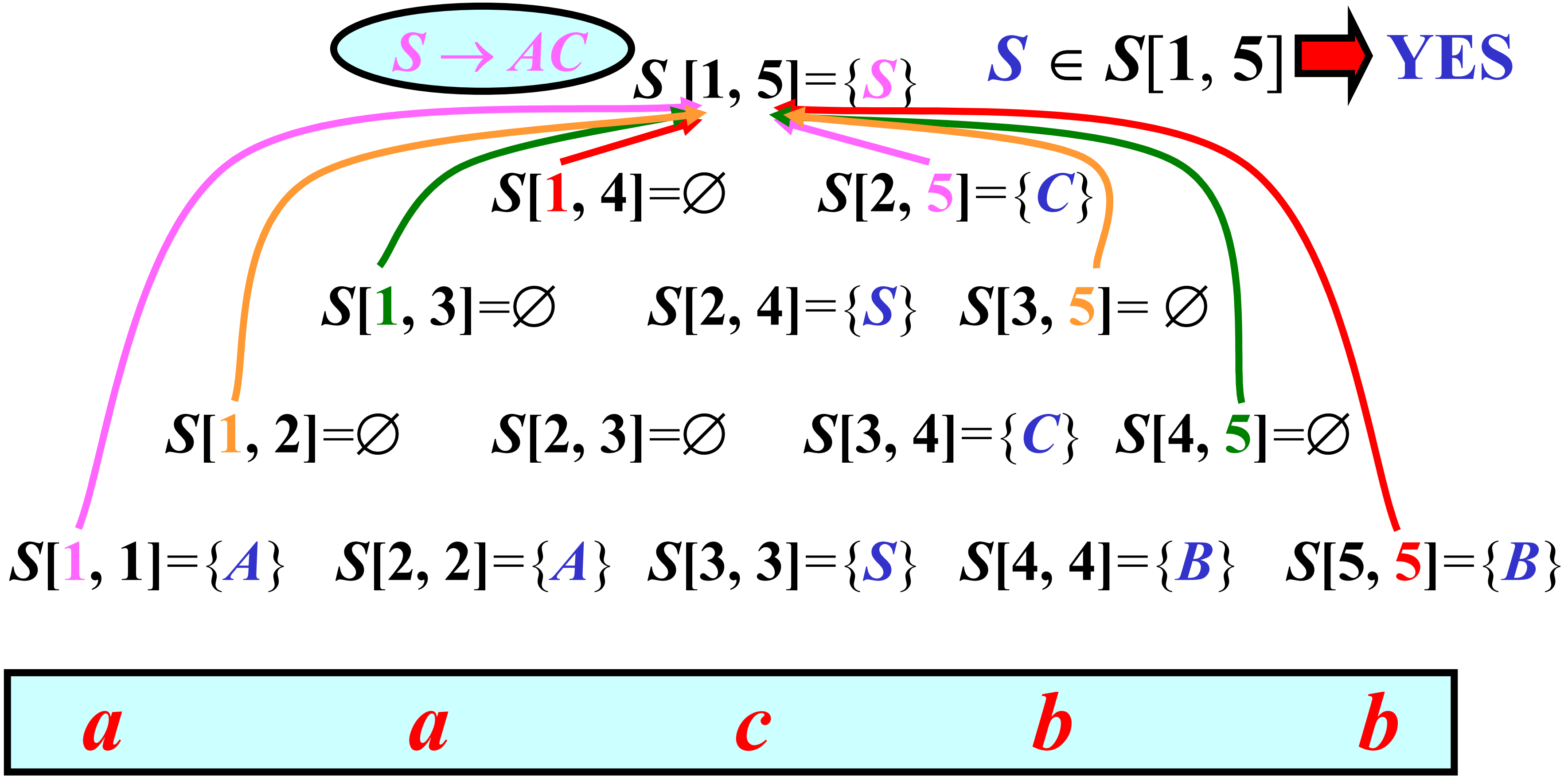
$S[1, 2]=\emptyset$ $S[2, 3]=\emptyset$ $S[3, 4]=\{C\}$ $S[4, 5]=\emptyset$

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

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

GP Based on CNF: Example 5/5

$G = (N, T, P, S)$, 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: $aacbb \in L(G)$?



$S \to AC$

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

$S \in S[1, 5]$ ⇒ YES

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

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

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

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

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 $m \geq 0$, $uv^mwx^my \in L$

Example:

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

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

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

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

$$\begin{array}{ccccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{b} & \mathbf{a} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ u & v & w & x & y \end{array}$$

$\mathbf{vx} = \mathbf{bb} \neq \varepsilon$

$|\mathbf{vwx}| = \mathbf{3}$: $\mathbf{1} \leq \mathbf{3} \leq \mathbf{5}$

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

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

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

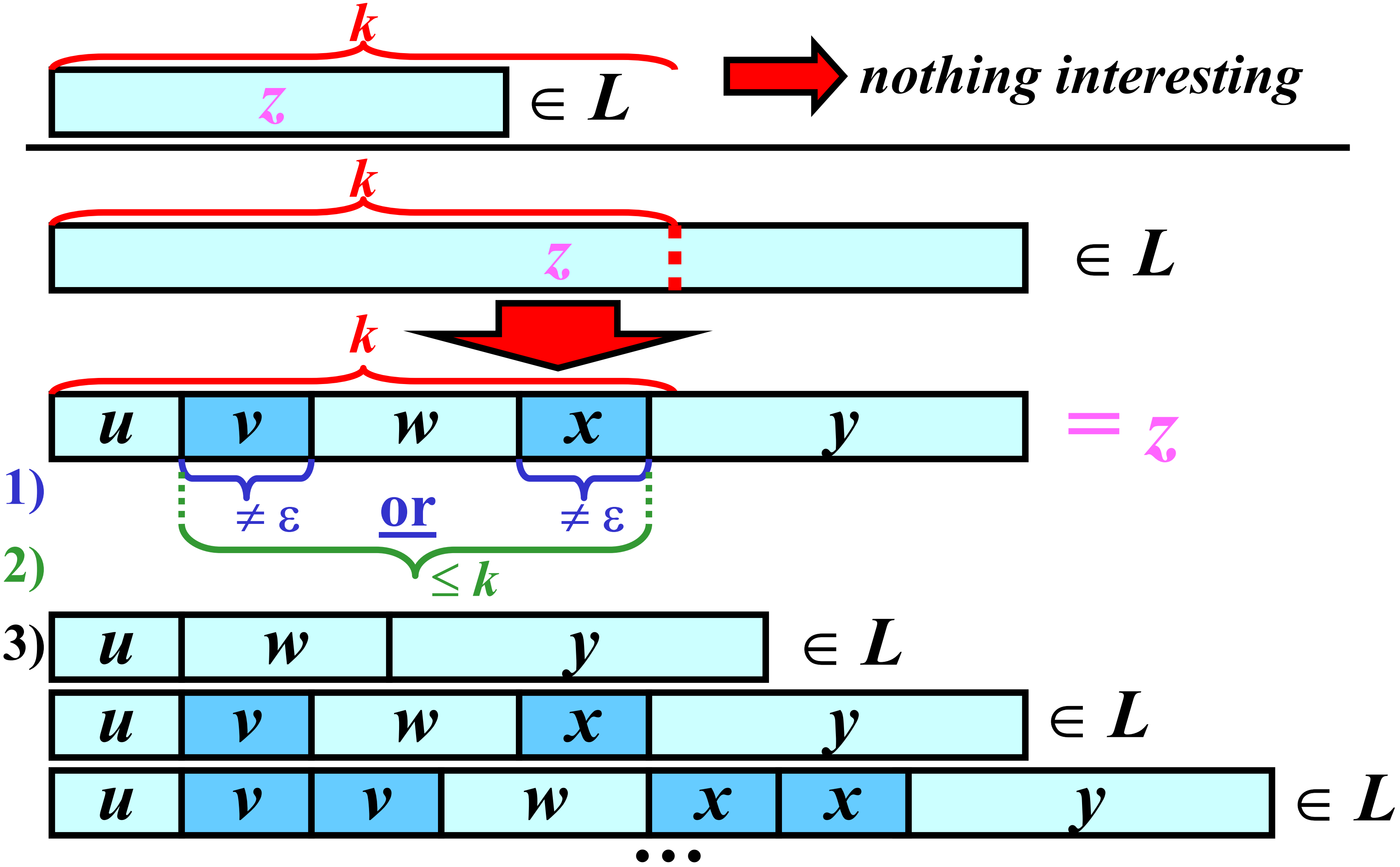
$\vdots$

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

$\vdots$

Pumping Lemma: Illustration

- L = any context-free language:



k

$z \in L$ → *nothing interesting*

k

$z \in L$

k

$uvwxy = z$

1) $v \neq \varepsilon$ **or** $x \neq \varepsilon$

2) $|vwx| \leq k$

3) $uwy \in L$

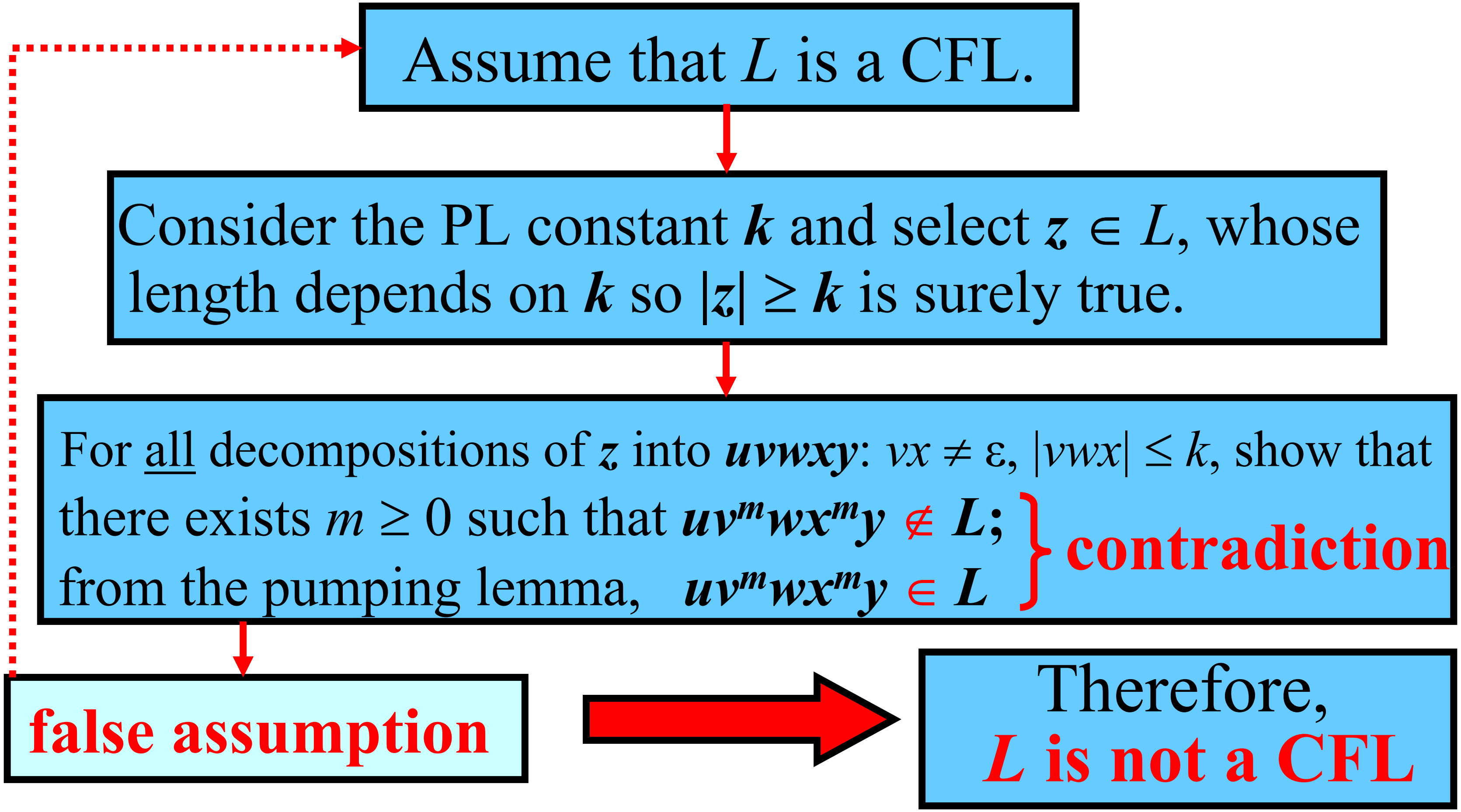
$uvwxy \in L$

$uvvwxxy \in L$

…

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 $|z| \geq \boldsymbol{k}$ is surely true.

For all decompositions of z into $\boldsymbol{uvwxy}$: $vx \neq \varepsilon$, $|vwx| \leq k$, show that there exists $m \geq 0$ such that $\boldsymbol{uv^m wx^m y} \notin \boldsymbol{L}$;

from the pumping lemma, $\boldsymbol{uv^m wx^m y} \in \boldsymbol{L}$

contradiction

false assumption

Therefore,

***L* is not a CFL**

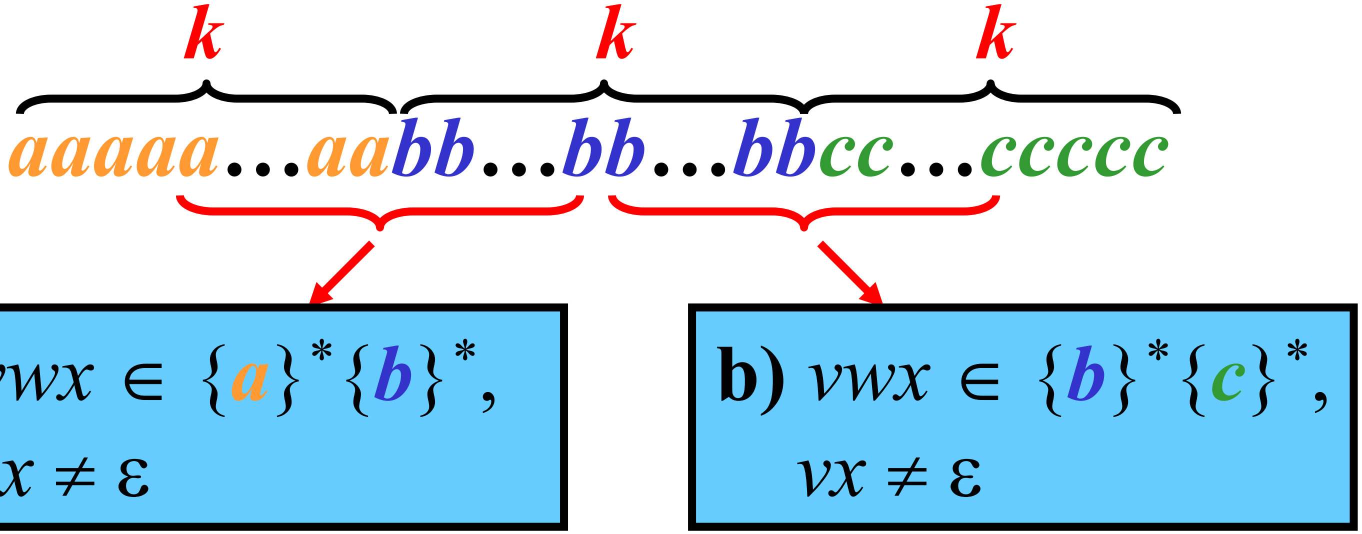
Pumping Lemma: Example 1/2

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

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

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

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

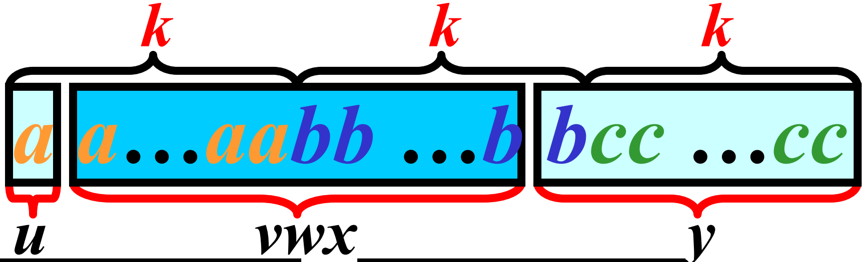


a) $vwx \in \{a\}^*\{b\}^*$, $vx \neq \varepsilon$

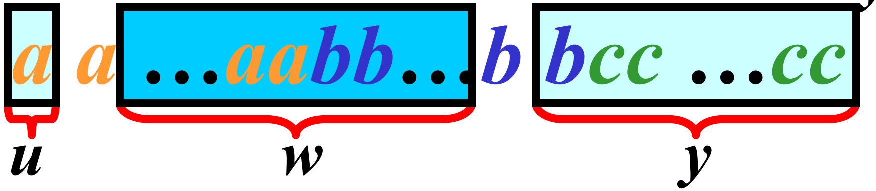
b) $vwx \in \{b\}^*\{c\}^*$, $vx \neq \varepsilon$

Pumping Lemma: Example 2/2

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



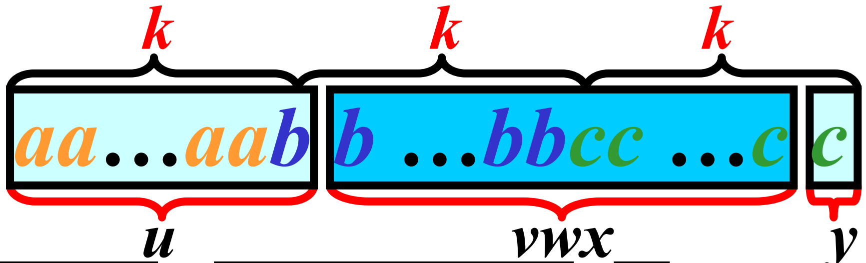
- Pumping lemma:
  $uv^0wx^0y \in L$
- $uv^0wx^0y = uwy =$



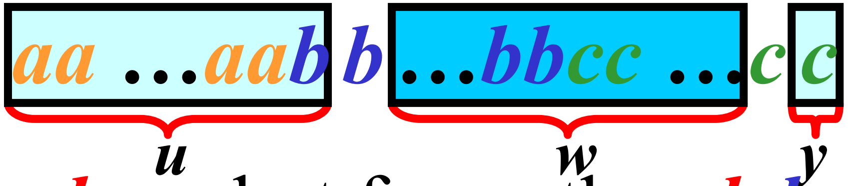
$\notin L$

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

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



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



$\notin L$

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

All these decompositions lead to a contradiction!

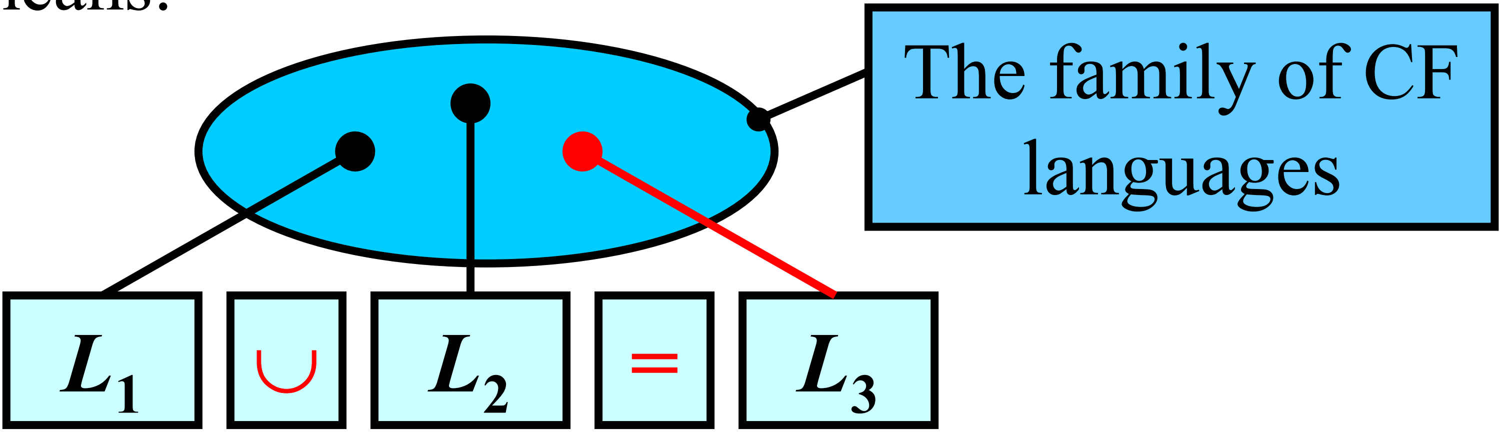
4) Therefore, L is not a CFL.

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:



The family of CF languages

$L_1 \cup L_2 = L_3$

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 $N_1 \cap N_2 = \emptyset$:
  - $N := \{S\} \cup N_1 \cup N_2$;
  - $P := \{S \rightarrow S_1, S \rightarrow S_2\} \cup P_1 \cup P_2$;

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 $N_1 \cap N_2 = \emptyset$:
  - $N := \{S\} \cup N_1 \cup N_2$;
  - $P := \{S \rightarrow S_1S_2\} \cup P_1 \cup P_2$;

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 $L(G_1) = L_1$, $L(G_2) = L_2$;
- 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:** $L(G) = \emptyset$?

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** $S \Rightarrow^{\boldsymbol{n}} w$, where $1 \leq \boldsymbol{n} \leq 2|w| - 1$, **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 $z \in L(M)$, $k \leq |z| < 2k$ **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.