Slide02 必做题

!Exercise 5.1.1（b）

参考解答: 从"课程文件"中下载网页文件，从中找到参考解答

Exercise 5.1.2 (c) 下面的文法产生了正则表达式 0*1(0+1)*的语言:

$$S \to A1B$$
$$A \to 0A \mid \varepsilon$$
$$B \to 0B \mid 1B \mid \varepsilon$$

试给出下列串的最左推导和最右推导:

c) 00011。

参考解答:

一个最左推导: $S \Rightarrow_{lm} A1B \Rightarrow_{lm} 0A1B \Rightarrow_{lm} 00A1B \Rightarrow_{lm} 000A1B \Rightarrow_{lm} 0001B$
$\Rightarrow_{lm} 00011B \Rightarrow_{lm} 00011$

一个最右推导: $S \Rightarrow_{rm} A1B \Rightarrow_{rm} A11B \Rightarrow_{rm} A11 \Rightarrow_{rm} 0A11 \Rightarrow_{rm} 00A11$
(rm 000A11 (rm 00011

Exercise 5.1.6(b) 如果有 $\alpha \overset{*}{\Rightarrow} \beta$ 和 $\beta \overset{*}{\Rightarrow} \gamma$，那么就有 $\alpha \overset{*}{\Rightarrow} \gamma$。提示: 使用归纳法，对推导 $\beta \overset{*}{\Rightarrow} \gamma$ 中的步数进行归纳。

参考解答: 归纳于推导 $\beta \overset{*}{\Rightarrow} \gamma$ 的步数.

基础 **步数为 0, 一定有 $\beta = \gamma$. 因为 $\alpha \overset{*}{\Rightarrow} \beta$**，所以 $\alpha \overset{*}{\Rightarrow} \gamma$。

归纳 **设步数大于 0，推导过程为: $\beta \Rightarrow^* \beta'$, $\beta' \Rightarrow \gamma$. 其中, $\beta \Rightarrow^* \beta'$ 的推导的步数少于推导 $\beta \overset{*}{\Rightarrow} \gamma$ 的步数, 根据归纳假设, 有 $\alpha \Rightarrow^* \beta'$ 成立. 这样,我们有 $\alpha \Rightarrow^* \beta'$ 和 $\beta' \Rightarrow \gamma$ 成立, 由定义可知 $\alpha \Rightarrow^* \gamma$.**

!!Exercise 5.1.8 考虑定义了下面的产生式的 CFG G:

$$S \to aSbS \mid bSaS \mid \varepsilon$$

证明 L(G)是所有有相同个数的 a 和 b 的串的集合。

参考解答:

应该证明: 一个串 *w* 中包含相同个数的 *a* 和 *b*, 当且仅当 $w \in L(G)$.

（$\Rightarrow$）假定 *w* 中包含相同个数的 *a* 和 *b*，现用数学归纳法通过对|*w*|进行归纳来证明 *w* 在 *L(G)* 中.

基础：长度 0 为归纳基础。如果$|w|$是 0，那么 w 一定是 ε，由于有产生式 $S\rightarrow\varepsilon$，，因此在有 $S\Rightarrow\varepsilon$，$\varepsilon\in L(G)$.

归纳：假定$|w|\geqslant 1$. 不妨设 w 的第一个字符为 a，因为 w 中包含相同个数的 a 和 b，所以总可以将 w 表示为 $w=aw_1bw_2$，其中 w_1 和 w_2 中都包含相同个数的 a 和 b，并满足：w_1 的任何前缀中，a 的个数不小于 b 的个数。由归纳假设，$S\Rightarrow^* w_1$，$S\Rightarrow^* w_2$，因此 $S\Rightarrow aSbS\Rightarrow^* aw_1bw_2$，所以 $w\in L(G)$.

（$\Leftarrow$）现在假定 $w\in L(G)$，即 $S\Rightarrow^* w$，要证明的是 w 中包含相同个数的 a 和 b. 证明的过程是对从 S 到 w 的推导过程的步数进行归纳。

基础：如果该推导是一步完成的，那么它一定使用了产生式 $S\rightarrow\varepsilon$，即 $w=\varepsilon$，显然 w 中包含相同个数（0 个）的 a 和 b.

归纳：现在，假定该推导共包含 $n+1$ 步，其中 $n\geqslant 1$，并且对于任何 n 步内完成的推导上述结论都成立——也就是说，如果 $S\Rightarrow^* x$ 可在 n 步内完成，那么 x 中包含相同个数的 a 和 b.

不妨设 w 的第一个字符为 a，考虑一个 w 的$(n+1)$步推导，它一定是如下形式：$S\Rightarrow aSbS\Rightarrow^* w$，

一定存在 w_1 和 w_2，满足 $w=aw_1bw_2$，且 $S\Rightarrow^* w_1$，$S\Rightarrow^* w_2$， 而这两个推导的步数都小于 n.

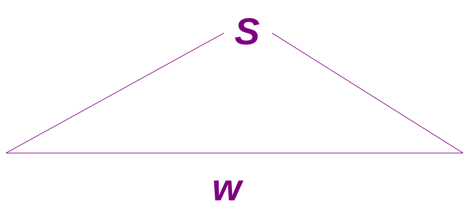
由归纳假设，w_1 和 w_2 中都包含相同个数的 a 和 b. 因此，w 中都包含相同个数的 a 和 b.

! Exercise 5.2.2 假设 G 是一个 CFG，并且它的任何一个产生式的右边都不是 ε。如果 w 在 L(G)中，w 的长度是 n，有一个 m 步完成的 w 的推导，证明有一个包含 n+m 个节点的关于 w 的分析树。

参考解答：

归纳于 $S\Rightarrow^* w$ 的步数 m. （S 为 G 的开始符号）

基础：$m=1$. 此时一定有产生式 $S\rightarrow w$，因此存在下图所示的 $n+1$ 个结点的分析树($w\neq\varepsilon$)，结果成立.



归纳：$m>1$. 设第一步使用了产生式 $S\rightarrow X_1X_2\ldots X_k$.

该推导如 $S\Rightarrow X_1X_2\ldots X_k\Rightarrow^* w$. 可以将 w 分成 $w=w_1w_2\ldots w_k$，其中

(a) 若 X_i 为终结符，则 $w_i=X_i$.

(b) 若 X_i 为非终结符，则 $X_i \Rightarrow^* w_i$. 且的步数 m_i 少于 m，由归纳假设，存在根结点为 X_i 的子分析树，其结点数为 $|w_i|+m_i$.

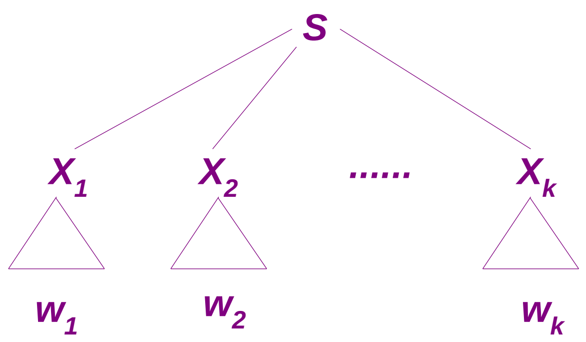
对于上述情形 ***(a)***，没有进一步的关于 X_i 的推导，可以认为 $m_i=0$. 这样，我们有如下关系：

$$m=m_1+m_2+\ldots+m_k+1$$

这样，存在一棵关于 w 的分析树（参见下图），其结点数为

$$1+(|w_1|+m_1)+(|w_2|+m_2)+\ldots+(|w_k|+m_k) = (|w_1|+|w_2|+\ldots+|w_k|)+(m_1+m_2+\ldots+m_k+1) = n+m$$

证毕.



Exercise 5.4.7(a)

参考解答:

由于该文法是无二义的，所以该串的最左、最右推导和分析树都是唯一的。

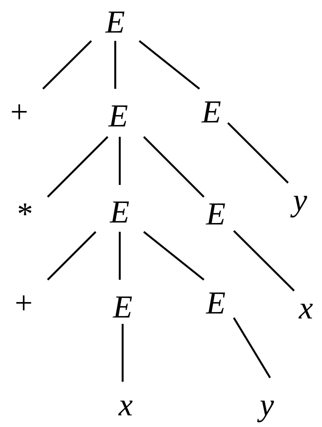
串 $+*-xyxy$ 的最左推导:

$$E\Rightarrow +EE\Rightarrow +*EEE\Rightarrow +*-EEEE\Rightarrow +*-xEEE\Rightarrow +*-xyEE\Rightarrow +*-xyxE\Rightarrow +*-xyxy$$

串 $+*-xyxy$ 的最右推导:

$$E \Rightarrow +EE \Rightarrow +Ey \Rightarrow +*EEy \Rightarrow +*Exy \Rightarrow +*-EExy \Rightarrow +*-Eyxy \Rightarrow +*-xyxy$$

串 $+*-xyxy$ 的分析树见下图:



附加 1 构造产生如下语言的上下文无关文法:

（1） $\{a^n b^{2n} c^m \mid n，m \geq 0\}$

（2） $\{a^n b^{n+m} c^m \mid n，m \geq 0\}$

（3） $\{ a^m b^n c^p d^q \mid m+n = p+q \}$

（4） $\{a^n b^i c^j d^m \mid n, m, i, j \geq 0 \wedge n + m = i + j \}$

（5） $\{ uawb \mid u,w \in \{a, b\}^* \wedge |u| = |w| \}$

参考解答:

（1）根据上下文无关文法的特点，要产生形如 $a^n b^{2n} c^m$ 的串，可以分别产生形如 $a^n b^{2n}$ 和形如 c^m 的串。设计好的文法是否就是该语言的文法？严格地说，应该给出证明。但若不是特别指明并且文法本身比较简单的话，通常可以忽略这一点（如果比较复杂的话应当给出思路的说明）。

对于该语言，存在一个由以下产生式定义的上下文无关文法 G[S]:

$$
\begin{aligned}
&S \rightarrow AB \\
&A \rightarrow \varepsilon \mid aAbb \\
&B \rightarrow \varepsilon \mid cB
\end{aligned}
$$

注：这里我们用 G[S] 表示文法 G 的开始符号为 S。

（2）根据上下文无关文法的特点，要产生形如 $a^n b^{n+m} c^m$ 的串，可以分别产生形如 $a^n b^n$ 和形如 $b^m c^m$ 的串。设计好的文法是否就是该语言的文法？严格地说，应该给出证明。但若不是特别指明并且文法本身比较简单的话，通常可以忽略这一点（如果比较复杂的话应当给出思路的说明）。

对于该语言，存在一个由以下产生式定义的上下文无关文法 G[S]:

$$
\begin{aligned}
&S \rightarrow AB \\
&A \rightarrow \varepsilon \mid aAb
\end{aligned}
$$

$$B \to \varepsilon \mid bBc$$

注：这里我们用 G[S] 表示文法 G 的开始符号为 S。

（3）我们可以通过“剥洋葱”的办法考虑：

i. 对于任何一个 L 中的串，如果是形如 $a^i w d^i$ 的形式，那么我们可以在两边剥掉相同个数的 a 和 d 直到不能再剥为止，剩下的肯定也是L中的串。

ii. 现在剩下的要么是 $a^j u c^j$ 的形式，要么是 $b^j v d^j$的形式，如果是前一种形式，那么我们在两边剥掉相同个数的 a 和 c 直到不能再剥为止；如果是后一种形式，那么我们在两边剥掉相同个数的 b 和 d 直到不能再剥为止。这两种情况最终剩下的 u 或者 v 都还是 L 中的串，而且只可能是 $b^k c^k$ 的形式。（请想想为什么不可能是 $a^k b^k$ 或者 $c^k d^k$ 的形式？）

iii. 明显，$b^k c^k$ 的形式的串可以通过产生式集合 $B \to bBc \mid \varepsilon$ 生成。

iv. 现在我们把这个“剥”的过程倒过来，把字母“包”回去，就可以获得以下的（本题的其中一种可能答案）：

$$\begin{aligned}
S &\to aSd \mid A \mid D \\
A &\to bAd \mid B \\
D &\to aDc \mid B \\
B &\to bBc \mid \varepsilon
\end{aligned}$$

（ 注：a 不多于 d 时，b 不少于 c；反之，a 不少于 d 时，b 不多于 c。前一种情形通过对应 A，后一种情形对应 D。）

（4）一个可能的上下文无关文法：

$$\begin{aligned}
S &\to A\,B \mid C\,D \\
A &\to a\,A\,b \mid \varepsilon \\
B &\to b\,B\,d \mid c\,E\,d \mid \varepsilon \\
E &\to c\,E\,d \mid \varepsilon \\
C &\to a\,C\,c \mid a\,A\,b \mid \varepsilon \\
D &\to c\,D\,d \mid \varepsilon
\end{aligned}$$

（ 注：在做有关文法设计的题目时，应尽可能训练少用非终结符。比如，对于此题，在某次期末试题中要求所使用的非终结符数目不超过 8）

（5）以下 G[S]是一种解法：

$$\begin{aligned}
S &\to Ab \\
A &\to BAB \mid a
\end{aligned}$$

$$B \to a \mid b$$

附加 2 给出语言 $\{ a^m b^n \mid m \geq 2n \geq 0\}$ 的二义文法和非二义文法各一个

参考解答:

可考虑分两个阶段：生成多余的 a，产生同样数目的 a 和 b：

$$S \to A \mid aS$$

$$A \to aaAb \mid \varepsilon$$

也可以考虑每次产生的时候要么在左右两侧分别加上 aa 和 b，要么只加上 a:

$$S \to aSb \mid aS \mid \varepsilon$$

$$S \to aaSb \mid aS \mid \varepsilon$$

前一个文法由于 aa 和 b 配对方式确定，因而是无二义的。以下是另一个无二义文法，大家可分析 aa 和 b 配对方式与前一个有何不同:

$$S \to aaSb \mid A$$
$$A \to Aa \mid \varepsilon$$

附加 3 适当变换文法，找到下列文法所定义语言的一个无二义的文法：

$$S \to SaS \mid SbS \mid ScS \mid d$$

参考解答:

该文法的形式很典型，可以先采用优先级联规则变换文法，然后再规定结合性对文法做进一步变换，即可消除二义性。

设 a、b 和 c 的优先级别依次增高，根据优先级联规则将文法变换为：

$$S \to SaS \mid A$$
$$A \to AbA \mid C$$
$$C \to CcC \mid d$$

规定结合性为左结合，进一步将文法变换为：

$$S \to SaD \mid D$$
$$D \to DbE \mid E$$
$$E \to EcF \mid F$$
$$F \to d$$

该文法为无二义的。

Slide02 思考题

!Exercise 5.1.1（c）

参考解答:

首先，奇数长度的串都不是 ww 的形式，这个容易处理。其次，偶数长度的串分成两个长度相等的段，不妨设每段的长度为 n; 因为不具有 ww 的形式，所以存在 $1 \le i \le n$，该串第 i 位和第 $n+i$ 位不同; 分别以第 i 位和第 $n+i$ 位为中心将该串重新划分为两段，长度分别为 $2(i-1)+1$ 和 $2(n-i)+1$; 这两段的中心不同，而围绕中心的其它位可以任意。

根据以上分析过程，如下产生式构成了 L 的一个上下文无关文法:

$$
\begin{aligned}
S &\to E \mid O \\
O &\to a \mid b \mid COC \\
E &\to AB \mid BA \\
A &\to CAC \mid a \\
B &\to CBC \mid b \\
C &\to a \mid b
\end{aligned}
$$

其中，开始符号为 S; 非终结符 O 负责产生奇数长度的串; 非终结符 E 负责产生偶数长度的串; 非终结符 A 负责产生以 a 为中心的串; 非终结符 B 负责产生以 b 为中心的串。

!Exercise 5.1.7（a）

参考解答:

对于 $w \in L(G)$，归纳于 $|w|$。

$|w| = 0$ 时，有 $w = \varepsilon$; 因为 G 中无 ε 产生式，所以 $w \notin L(G)$。

$|w| = 1$ 时，要使得 $w \notin L(G)$，只有使用产生式 $S\to a$ 或 $S\to b$，所以 $w = a$ 或 $w = b$; w 中没有子串 ba。

当$|w| > 1$ 时，第一步推导必定使用产生式 $S \to aS$ 或 $S \to Sb$，而在随后的推导步中从 S 出发可推导出 $w' \in L(G)$，并且$|w'|$ 小于$|w|$; 根据归纳假设，w'中没有子串 ba; 由于 $w= a w'$ 或 $w= w'b$，所以 w 中也没有子串 ba。

Exercise 5.4.7(b)

参考解答:

先证明对任何终结符串 w，如下命题成立:

命题 P: $E \Rightarrow^* w$, iff w 中 x 和 y 的总数比+，*和–的总数多 1，并且 w 的任何真前缀中+，*和–的总数不少于 x 和 y 的总数

（only if）归纳于 $E \Rightarrow^* w$ 的步数 k。

若 k=1，即 $E \Rightarrow w$ ，则必有 $w=x$ 或 $w=y$，only if 部分的条件成立。

若 k >1，则推导的第一步一定使用了三个产生式 $E \to +EE \mid *EE \mid -EE$ 之一，不妨设使用了产生式 $E \to +EE$。此时，w 可表示为 $w = +w_1w_2$ ，并且有 $E \Rightarrow^* w_1$ 和 $E \Rightarrow^* w_2$ 。而 $E \Rightarrow^* w_1$ 和 $E \Rightarrow^* w_2$ 的推导步数均小于 k，所以 w_1 和 w_2 中 x

和 y 的总数比+，*和–的总数多 1，且其中的任何真前缀中+，*和–的总数不少于 x 和 y 的总数。这样，我们可以推出：$w = + w_1w_2$ ，中 x 和 y 的总数比+，*和–的总数多 1，且其中的任何真前缀中+，*和–的总数不少于 x 和 y 的总数。

（if）归纳于 w 的长度 k。

若 k=1，因 w 中 x 和 y 的总数比+，*和–的总数多 1，必有 $w=x$ 或 $w=y$，所以 $E \Rightarrow^* w$ 成立。

若 k >1，因 w 的任何真前缀中+，*和–的总数不少于 x 和 y 的总数，所以 w 的第一个符号是 +，* 或 – 之一，不妨设为+。我们将 w 表示为 $w = + w_1w_2$ ，这里 $+ w_1$ 是首次满足 w_1 中 x 和 y 的总数比+，*和–的总数多 1 的 w 的真前缀（由于 w 中 x 和 y 的总数比+，* 和 – 的总数多 1，这样的非空子串 w_1 和 w_2 总是可以找到的）。不难推断：w_1 和 w_2 都满足：x 和 y 的总数比+，*和–的总数多 1，且其中的任何真前缀中+，*和–的总数不少于 x 和 y 的总数。依归纳假设，我们有 $E \Rightarrow^* w_1$ 和 $E \Rightarrow^* w_2$ 。因此，$E \Rightarrow^* w$ 成立。

下面证明文法的无二义性，即证明对所有终结符串，其分析树或最左推导是唯一的。对于本题，可以采取的办法是归纳于终结符串的长度，以证明该文法所产生的任何终结符串的最左推导是唯一的。

设 w 表示该文法可推导出的任何字符串，现归纳于 w 的长度来证明其最左推导是唯一的。

基础：$|w|=1$ 时，必有 $w=x$ 或 $w=y$；其最左推导是 $E \Rightarrow x$ 或 $E \Rightarrow y$，是唯一的。

归纳：设 $|w| < k\ (k >1)$ 时，w 有唯一的最左推导。当 $|w| = k$ 时，产生 w 的第一步推导一定使用了三个产生式 $E \to +EE \mid *EE \mid -EE$ 之一（因为 $k >1$）；不妨设 w 的第一个符号为 +，则第一步推导是唯一的，只能是 $E \Rightarrow +EE$；根据上下文无关文法的特性，存在 w_1、w_2 ，满足 $w = + w_1w_2$（根据上述命题 P，这样 w_1 和 w_2 的是唯一确定的，想想为什么？），并且有 $E \Rightarrow^* w_1$ 和 $E \Rightarrow^* w_2$ ；因为 $|w_1| < k$ 以及 $|w_2| < k$，根据归纳假设，w_1 和 w_2 的最左推导是唯一的；连同唯一的第一步推导，就可得到一个 w 的最左推导，且是唯一的最左推导。

因此，该文法是无二义的。

附加：（1） 设 G 为上下文无关文法，其终结符集合为 $\{a，b，c\}$，开始符号为 S，产生式集合如下：

$$S \to A \mid aSc$$
$$A \to B \mid bAc$$
$$B \to \varepsilon \mid Bc$$

试证明 $L(G) = \{ a^i b^j c^k \mid i + j \le k$，其中 $i、j、k$ 均为自然数$\}$。

证明： 可先后证明下列命题（注意这里并不需要互归纳）：

1）对任何 w，$B \Rightarrow^* w$ 当且仅当 $w \in \{ c^k \mid k \ge 0 \}$；

2）对任何 w，$A \Rightarrow^* w$ 当且仅当 $w \in \{ b^j c^k \mid j \ge 0，k \ge 0，j \le k \}$；

3）对任何 w，$S \Rightarrow^* w$ 当且仅当 $w \in \{a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i+j \leq k\}$;

命题 1）的证明:

先证对任何 w，如果 $B \Rightarrow^* w$， 则 $w \in \{c^k \mid k \geq 0\}$。归纳于 $B \Rightarrow^* w$ 的推导步数 n。

基础: n=1 时，必有 w=$\varepsilon \in \{c^k \mid k \geq 0\}$。

归纳: 假设 $n < m$ 时，$w \in \{c^k \mid k \geq 0\}$。当 $n = m$ 时，第一步推导必然使用了产生式 $B \to Bc$，则有 $w= w'c$ 和 $B \Rightarrow^* w'$，且 $B \Rightarrow^* w'$的推导步数小于 m; 根据归纳假设，$w' \in \{c^k \mid k \geq 0\}$，那么有 $w= w'c \in \{c^k \mid k \geq 0\}$。

再证对任何 $w\in \{c^k \mid k \geq 0\}$，$B \Rightarrow^* w$。归纳于 w 的长度 $|w|$。

基础: $|w|$=0 时，必有 w=ε ; 使用产生式 $B \to \varepsilon$ 一次，可以得出 $B \Rightarrow^* w$。

归纳: 假设$|w| < n$ 时，$B \Rightarrow^* w$ 成立。当$|w| = n$ 时，可令 w=$w'c$，其中 w'满足 $|w'| < n$; 根据归纳假设，有推导 $B \Rightarrow^* w'$; 又因有直接推导 $B \Rightarrow Bc$，故有推导 $B \Rightarrow^* w$。

命题 2）的证明:

先证对任何 w，如果 $A \Rightarrow^* w$， 则 $w \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$。归纳于 $A \Rightarrow^* w$ 的推导步数 n。

基础: n=2 时（n 不可能为 1），必有 w=$\varepsilon \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$。

归纳: 假设 $n < m$ 时，$w \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$。当 $n = m$ 时，第一步推导或者使用产生式 $A \to B$，或者使用产生式 $A \to bAc$。第一步推导若是使用了产生式 $A \to B$，则有 $B \Rightarrow^* w$; 根据命题 1)，有 $w \in \{c^k \mid k \geq 0\}$，自然也有 $w \in\{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$。第一步推导若是使用了产生式 $A \to bAc$，则有 w=$bw'c$ 和 $A \Rightarrow^* w'$，且 $A \Rightarrow^* w'$的推导步数小于 m; 根据归纳假设，$w' \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$，那么也有 $w= bw'c \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$。

再证对任何 $w\in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$，$A \Rightarrow^* w$。归纳于 w 的长度 $|w|$。

基础: $|w|$=0 时，必有 w=ε ; 使用直接推导 $A \Rightarrow B$ 和 $B \Rightarrow^* \varepsilon$ （由命题 1)），可以得出 $A \Rightarrow^* w$。

归纳: 假设$|w| < n$ 时，$A \Rightarrow^* w$ 成立。当$|w| = n$ 时，可令 w=$bw'c$（w 中 b 的数目不等于 0），或 $w = w''c$（w 中 b 的数目等于 0）。对于前者，显然有 $w' \in \{b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k\}$，且满足 $|w'| < n$; 根据归纳假设，有推导 $A \Rightarrow^* w'$; 因有直接推导 $A \Rightarrow bAc$，故有推导 $A \Rightarrow^* bw'c = w$。对于后者，显然有 $w'' \in \{c^k \mid k \geq 0\}$，由命题 1）可知 $B \Rightarrow w''$; 因有直接推导 $A \Rightarrow B$，$B \Rightarrow Bc$ 和 $B \Rightarrow \varepsilon$，故有推导 $A \Rightarrow^* w''c = w$。

命题 3）的证明:

先证对任何 w，如果 $S \Rightarrow^* w$， 则 $w \in \{a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i+j \leq k\}$。归纳于 $S \Rightarrow^* w$ 的推导步数 n。

基础：n=3 时（n 不可能为 1，2），必有 $w=\varepsilon \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$。

归纳：假设 $n < m$ 时，$w \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$。当 $n = m$ 时，第一步推导或者使用产生式 $S \to A$，或者使用产生式 $S \to aSc$。第一步推导若是使用了产生式 $S \to A$，则有 $A \Rightarrow^* w$；根据命题 2），有 $w \in \{ b^j c^k \mid j \geq 0,\ k \geq 0,\ j \leq k \}$，自然也有 $w \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$。第一步推导若是使用了产生式 $S \to aSc$，则有 $w=aw'c$ 和 $S \Rightarrow^* w'$，且 $S \Rightarrow^* w'$的推导步数小于 m；根据归纳假设，$w' \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$，那么也有 $w= aw'c \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$。

再证对任何 $w \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$，$S \Rightarrow^* w$。 归纳于 w 的长度 $|w|$。

基础：$|w|=0$ 时，必有 $w=\varepsilon$ ；使用直接推导 $S \Rightarrow A$ 和 $A \Rightarrow^* \varepsilon$ （由命题 2）），可以得出 $S \Rightarrow^* w$。

归纳：假设$|w| < n$ 时，$S \Rightarrow^* w$ 成立。当$|w| = n$ 时，可令 $w=aw'c$（w 中 a 的数目不等于 0），或 $w=bw''c$（w 中 a 的数目等于 0）。对于前者，显然有 $w' \in \{ a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k \geq 0,\ i + j \leq k \}$，且满足 $|w'| < n$；根据归纳假设，有推导 $S \Rightarrow^* w'$；因有直接推导 $S \Rightarrow aSc$，故有推导 $S \Rightarrow^* aw'c = w$。对于后者，显然有 $w'' \in \{ b^j c^k \mid j \geq 0,\ k \geq 0,\ i + j \leq k \}$，由命题 2）可知 $A \Rightarrow w''$；因有直接推导 $S \Rightarrow A$，$A \Rightarrow bAc$，故有推导 $S \Rightarrow^* bw''c = w$。

附加：（2）设 G 为上下文无关文法，其终结符集合为 {a, b}，开始符号为 S，产生式集合如下：

$$
\begin{aligned}
&S \to \varepsilon \mid aB \mid bA \\
&A \to a \mid aS \mid bAA \\
&B \to b \mid bS \mid aBB
\end{aligned}
$$

试证明 $L(G) = \{ w \mid w \in \{a, b\}^*, occur(w, a) = occur(w, b) \}$.
其中，对于符号 a 和串 w，occur(a, w) 表示 a 在 w 中出现的次数.

证明：可证明对所有的 $w \in \{a, b\}^*$，有如下三个等价式成立（互归纳）：

1）S =>* w iff occur(a, w) =occur(b，w)；
2）A =>* w iff $|w|>0 \wedge$ occur(a, w) = occur(b，w) + 1；
3）B =>* w iff $|w|>0 \wedge$ occur(b, w) = occur(a，w) + 1

1） 成立时即题设成立

为方便，可以将 (if) 和（only if）分开证明。

首先，我们归纳于三个式子中 =>*的步数（统一用 n 表示）， 用互归纳方法证明：对所有的 $w \in \{a, b\}^+$，

1）if S =>* w , then occur(a, w) =occur(b，w)；

2）if $A \Rightarrow^* w$, then $|w|>0 \wedge occur(a, w) = occur(b, w) + 1$；

3）if $B \Rightarrow^* w$, then $|w|>0 \wedge occur(b, w) = occur(a, w) + 1$

(基础) 1）对于 $S \Rightarrow^* w$。

当 $n=1$ 时，由 $S \Rightarrow w$ 知 $w = \varepsilon$，显然有 $occur(a, w) = occur(b, w) = 0$。

2）对于 $A \Rightarrow^* w$。

当 $n=1$ 时，由 $A \Rightarrow w$ 知 $w = a$，显然有$|w|>0 \wedge occur(a, w) = occur(b, w) + 1$。

3）对于 $A \Rightarrow^* w$。

当 $n=1$ 时，由 $B \Rightarrow w$ 知 $w = b$，显然有 $|w|>0 \wedge occur(b, w) = occur(a, w) + 1$。

(归纳) 1）当 $n>1$ 时，$S \Rightarrow^* w$ 第一步必使用产生式 $S \to aB$ 或 $S \to bA$。

若使用产生式 $S \to aB$，则推导过程为 $S \Rightarrow aB \Rightarrow^* aw'$；此时，我们有 $B \Rightarrow^* w'$ 的推导步数小于 n，根据归纳假设，有 $|w'|>0 \wedge occur(b, w') = occur(a, w') + 1$，因此 $w = aw'$ 满足 $occur(a, w) = occur(b, w)$。

若 $S \Rightarrow^* w$ 第一步使用产生式 $S \to bA$，则推导过程为 $S \Rightarrow bA \Rightarrow^* bw'$；此时，我们有 $A \Rightarrow^* w'$的推导步数小于 n，根据归纳假设，有 $|w'|>0 \wedge occur(a, w') = occur(b, w') + 1$，因此 $w = bw'$ 满足 $occur(a, w) = occur(b, w)$。

2）当 $n>1$ 时，$A \Rightarrow^* w$ 第一步必使用产生式 $A \to aS$ 或 $A \to bAA$。

若使用产生式 $A \to aS$，则推导过程为 $A \Rightarrow aS \Rightarrow^* aw'$；此时，我们有 $S \Rightarrow^* w'$ 的推导步数小于 n，根据归纳假设，有 $occur(b, w') = occur(a, w')$，因此 $w = aw'$ 满足 $|w|>0 \wedge occur(a, w) = occur(b, w) + 1$。

若 $A \Rightarrow^* w$ 第一步使用产生式 $A \to bAA$，则推导过程为 $A \Rightarrow bAA \Rightarrow^* bw'w''$，这里，我们有 $A \Rightarrow^* w'$ 和 $A \Rightarrow^* w''$ 的推导步数均小于 n，根据归纳假设，有 $|w'|>0 \wedge occur(a, w') = occur(b, w') + 1$ 和 $|w''|>0 \wedge occur(a, w'') = occur(b, w'') + 1$，因此 $w = bw'w''$ 满足 $|w|>0 \wedge occur(a, w) = occur(b, w) + 1$。

3）当 $n>1$ 时，$B \Rightarrow^* w$ 第一步必使用产生式 $B \to bS$ 或 $B \to aBB$。

若使用产生式 $B \to bS$，则推导过程为 $B \Rightarrow bS \Rightarrow^* bw'$；此时，我们有 $S \Rightarrow^* w'$ 的推导步数小于 n，根据归纳假设，有 $occur(b, w') = occur(a, w')$，因此 $w = bw'$ 满足 $|w|>0 \wedge occur(b, w) = occur(a, w) + 1$。

若 $B \Rightarrow^* w$ 第一步使用产生式 $B \to aBB$，则推导过程为 $B \Rightarrow aBB \Rightarrow^* aw'w''$，这里，我们有 $B \Rightarrow^* w'$ 和 $B \Rightarrow^* w''$ 的推导步数均小于 n，根据归纳假设，有 $|w'|>0 \wedge occur(b, w') = occur(a, w') + 1$ 和 $|w''|>0 \wedge occur(b, w'') = occur(a, w'') + 1$，因此 $w = aw'w''$ 满足 $|w|>0 \wedge occur(b, w) = occur(a, w) + 1$。

其次，我们归纳于 $|w|$，用互归纳方法证明：对所有的 $w \in \{a, b\}^+$，

1）if $occur(a, w) = occur(b, w)$, then $S \Rightarrow^* w$；

2）if $|w|>0 \wedge occur(a, w) = occur(b, w) + 1$, then $A \Rightarrow^* w$；

3）if $|w|>0 \wedge occur(b, w) = occur(a, w) + 1$, then $B \Rightarrow^* w$

(基础) 当 $|w| =0$ 时，即 $w = \varepsilon$.

1）前提 $occur(a, w) = occur(b, w)$ 成立，结论 $S \Rightarrow^* w$ 也成立；

2）前提不成立；

3）前提不成立

(归纳) 当 $|w| >0$ 时，即存在 w’ 满足 w = aw’ 或 w = bw’.

1） 设 occur(a, w) =occur(b，w)。

若 w = aw’，则有 $|w'|>0 \wedge$ occur(b, w’) = occur(a，w’) + 1。由归纳假设，B =>* w’，所以，有 S => aB =>* aw’ = w.

若 w = bw’，则有 $|w'|>0 \wedge$ occur(a, w’) = occur(b，w’) + 1。由归纳假设，A =>* w’，所以，有 S => bA =>* bw’ = w.

2） 设 $|w|>0 \wedge$ occur(a, w) = occur(b，w) + 1。

若 w = aw’，则有 occur(a, w’) =occur(b，w’)。由归纳假设，S =>* w’，所以，有 A => aS =>* aw’ = w.

若 w = bw’，则有 $|w'|>1 \wedge$ occur(a, w’) =occur(b，w’)+2。此时，必存在 w'_1 和 w'_2，满足 $w'= w'_1 w'_2$ ，以及 $|w'_1|>0 \wedge$ occur(a, w'_1) =occur(b, w'_1)+1 和 $|w'_2|>0 \wedge$ occur(a, w'_2) =occur(b, w'_2)+1。由归纳假设，我们有 A =>* w'_1 和 A =>* w'_2，所以，有 A => bAA =>* b $w'_1 w'_2$ = bw’ = w.

3） 设 $|w|>0 \wedge$ occur(b, w) = occur(a，w) + 1。

若 w = aw’，则有 $|w'|>1 \wedge$ occur(b, w’) =occur(a，w’)+2。此时，必存在 w'_1 和 w'_2，满足 $w'= w'_1 w'_2$ ，以及 $|w'_1|>0 \wedge$ occur(b, w'_1) =occur(a, w'_1)+1 和 $|w'_2|>0 \wedge$ occur(b, w'_2) =occur(a, w'_2)+1。由归纳假设，我们有 B =>* w'_1 和 B =>* w'_2，所以，有 B => aBB =>* a$w'_1 w'_2$ = aw’ = w.

若 w = bw’，则有 occur(b, w’) =occur(a，w’)。由归纳假设，S =>* w’，所以，有 B => bS =>* bw’ = w.

证毕。