Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:
For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \rightarrow \beta$ indicates that you can substitute β for α.

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:

For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \to \beta$ indicates that you can substitute β for α.
If, when using a production, we start with string γ_1 and end up with γ_2, we denote this by $\gamma_1 \Rightarrow \gamma_2$

Example:
$\Phi = \{a, b, A, B\}$,
$P = \{ab \to AB; bab \to a; A \to bb\}$

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:

For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \to \beta$ indicates that you can substitute β for α.
If, when using a production, we start with string γ_1 and end up with γ_2, we denote this by $\gamma_1 \Rightarrow \gamma_2$

Example:
$\Phi = \{a, b, A, B\}$,
$P = \{ab \to AB; bab \to a; A \to bb\}$

Then, $ababAA \Rightarrow aaAA \Rightarrow aabbA \Rightarrow aABbA$

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:
For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \to \beta$ indicates that you can substitute β for α.
If, when using a production, we start with string γ_1 and end up with γ_2, we denote this by $\gamma_1 \Rightarrow \gamma_2$

Example:
$\Phi = \{a, b, A, B\}$,
$P = \{ab \to AB; bab \to a; A \to bb\}$

Then, $ababAA \Rightarrow aaAA \Rightarrow aabbA \Rightarrow aABbA$
We will sometimes write: $ababAA \overset{*}{\Rightarrow} aABbA$

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:
For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \rightarrow \beta$ indicates that you can substitute β for α.
If, when using a production, we start with string γ_1 and end up with γ_2, we denote this by $\gamma_1 \Rightarrow \gamma_2$
S: special "start" symbol, $S \in V$.

Grammars

A grammar G is defined by a quadruple: $G = \{V, \Sigma, S, P\}$
Σ: is the alphabet. G generates strings over Σ.
V: a set of symbols, disjoint from Σ: $\Sigma \cap V = \emptyset$
P: a set of *productions*, which are substitutions rules:
For $\alpha, \beta \in \Phi$, $\alpha \neq \Lambda$, $\alpha \rightarrow \beta$ indicates that you can substitute β for α.
If, when using a production, we start with string γ_1 and end up with γ_2, we denote this by $\gamma_1 \Rightarrow \gamma_2$
S: special "start" symbol, $S \in V$.

The language generated by G, $\mathcal{L}(G) = \{x \mid S \overset{*}{\Rightarrow} x \wedge x \in \Sigma^*\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

Grammar Examples

Example:

Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$

$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$

$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language?

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\big\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\big\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow aC;\ S \Rightarrow bC;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\big\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \left\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\right\}$
$P = \left\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\right\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \left\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\right\}$
$P = \left\{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\right\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\big\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \big\{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\big\}$
$P = \big\{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\big\}$
$V = \{S, C\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow \Lambda;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow \Lambda;\ S \Rightarrow aS;$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aA, S \to \Lambda, A \to bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow \Lambda;\ S \Rightarrow aS;\ S \Rightarrow bcS;\}$

Grammar Examples

Example:
Let $G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aA, S \rightarrow \Lambda, A \rightarrow bS\})$
$S \Rightarrow aA \Rightarrow abS \Rightarrow ab$
$S \Rightarrow aA \Rightarrow abS \Rightarrow abaA \Rightarrow ababS \Rightarrow abab$

What is the language? It is: $(ab)^*$

Give a grammar for RE $r = a + bc^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow a;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + b)c^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow aC;\ S \Rightarrow bC;\ C \Rightarrow cC;\ C \Rightarrow \Lambda\}$
$V = \{S, C\}$

Give a grammar for RE $r = (a + bc)^*$
$G = \{\{S, ?\}, \{a, b, c\}, S, P = \{?\}\}$
$P = \{S \Rightarrow \Lambda;\ S \Rightarrow aS;\ S \Rightarrow bcS;\}$
$V = \{S\}$

Connection to Natural Language

$$\langle sentence \rangle \rightarrow \langle noun - phrase \rangle \langle verb - phrase \rangle$$
$$\langle noun - phrase \rangle \rightarrow \langle article \rangle \langle noun \rangle$$
$$\langle noun - phrase \rangle \langle auxillary \rangle \rightarrow \langle auxillary \rangle \langle noun - phrase \rangle$$

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \rightarrow \Lambda$ or $A \rightarrow bC$, where $A, C \in V$ and $b \in \Sigma^*$

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \rightarrow \Lambda$ or $A \rightarrow bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \stackrel{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \stackrel{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Proof by induction on k, the length of the derivation $S \stackrel{*}{\Rightarrow} x$.

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \stackrel{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Proof by induction on k, the length of the derivation $S \stackrel{*}{\Rightarrow} x$.
If $k = 0$, then $x = S, y = \Lambda, B = S$

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \stackrel{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Proof by induction on k, the length of the derivation $S \stackrel{*}{\Rightarrow} x$.
If $k = 0$, then $x = S, y = \Lambda, B = S$
Suppose the claim holds for a derivation using k production rules.
A derivation of length $k + 1$ must have been of the form $S \stackrel{*}{\Rightarrow} yB$ after k steps.

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \overset{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Proof by induction on k, the length of the derivation $S \overset{*}{\Rightarrow} x$.
If $k = 0$, then $x = S, y = \Lambda, B = S$
Suppose the claim holds for a derivation using k production rules.
A derivation of length $k + 1$ must have been of the form $S \overset{*}{\Rightarrow} yB$ after k steps.
For the last production, we either apply the rule:
$B \to \Lambda$, in which case, $S \overset{*}{\Rightarrow} y$, with $y \in \Sigma^*$, or we apply

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \stackrel{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Proof by induction on k, the length of the derivation $S \stackrel{*}{\Rightarrow} x$.
If $k = 0$, then $x = S, y = \Lambda, B = S$
Suppose the claim holds for a derivation using k production rules.
A derivation of length $k + 1$ must have been of the form $S \stackrel{*}{\Rightarrow} yB$ after k steps.
For the last production, we either apply the rule:
$B \to \Lambda$, in which case, $S \stackrel{*}{\Rightarrow} y$, with $y \in \Sigma^*$, or we apply
$B \to aC$, in which case we have $S \stackrel{*}{\Rightarrow} yaC$ for $ya \in \Sigma^*$ and $C \in V$.

Regular Grammars

Regular grammars are the same, but with a restriction on the production rules.
2 types of rule are allowed:
$A \to \Lambda$ or $A \to bC$, where $A, C \in V$ and $b \in \Sigma^*$

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.1

For any regular grammar $G = (V, \Sigma, S, P)$, if $S \overset{*}{\Rightarrow} x$, then either $x \in \Sigma^*$, or $x = yB$, where $y \in \Sigma^*$ and $B \in V$.

Definition

A *unit production* is a production rule of the form $A \to B$, where $A, B \in V$.
A regular grammar with unit productions is a regular grammar that also allows unit production rules.

Structure of proof of Theorem 8.1

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

Structure of proof of Theorem 8.1

Theorem 8.1

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G such that $\mathcal{L}(G) = L$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

Lemma 8.3

If $L = \mathcal{L}(G)$ for some regular grammar with unit productions, then there exists a regular grammar G' such that $L = \mathcal{L}(G')$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

Algorithm 8.1

Constructing a regular grammar from an RE
Input: A RE r such that $\mathcal{L}(r) \subseteq \Sigma^*$.
Output: A regular grammar $G = (V, \Sigma, S, P)$ with unit productions, $\mathcal{L}(r) = \mathcal{L}(G)$

if r has no operators **then**
  if $r = a$ **then return** $(\{S, A\}, \Sigma, S, \{S \to aA, A \to \Lambda\})$
  if $r = \Lambda$ **then return** $(\{S\}, \Sigma, S, \{S \to \Lambda\})$
  if $r = \emptyset$ **then return** $(\{S\}, \Sigma, S, \emptyset)$
else
  if $r = r_1 + r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$
  if $r = r_1 r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    for each $A \to \Lambda \in P_1$ **do**
      replace $A \to \Lambda$ by $A \to S_2$ in P_1
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$
  if $r = r_1^*$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    for each $A \to \Lambda \in P_1$ **do**
      replace $A \to \Lambda$ by $A \to S$ in P_1
    return $(V_1 \cup \{S\}, \Sigma, S, P_1 \cup \{S \to \Lambda, S \to S_1\})$

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow \text{Algorithm 8.1 } (r_1)$
$(V_2, \Sigma, S_2, P_2) \Leftarrow \text{Algorithm 8.1 } (r_2)$
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.
Since $x \in \mathcal{L}(r)$, we have $x \in \mathcal{L}(r_1) \lor x \in \mathcal{L}(r_2)$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
$(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.
Since $x \in \mathcal{L}(r)$, we have $x \in \mathcal{L}(r_1) \vee x \in \mathcal{L}(r_2)$.
By the strong inductive hypothesis, if $x \in \mathcal{L}(r_1)$, then it can be generated using grammar (V_1, Σ, S_1, P_1).

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.
Since $x \in \mathcal{L}(r)$, we have $x \in \mathcal{L}(r_1) \vee x \in \mathcal{L}(r_2)$.
By the strong inductive hypothesis, if $x \in \mathcal{L}(r_1)$, then it can be generated using grammar (V_1, Σ, S_1, P_1).
Let $S_1 \stackrel{*}{\Rightarrow} x$ be the sequence of productions that generates x in that grammar.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
$(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.
Since $x \in \mathcal{L}(r)$, we have $x \in \mathcal{L}(r_1) \vee x \in \mathcal{L}(r_2)$.
By the strong inductive hypothesis, if $x \in \mathcal{L}(r_1)$, then it can be generated using grammar (V_1, Σ, S_1, P_1).
Let $S_1 \stackrel{*}{\Rightarrow} x$ be the sequence of productions that generates x in that grammar.
Then $S \Rightarrow S_1 \stackrel{*}{\Rightarrow} x$ is a valid sequence in grammar G.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 + r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
$(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1,\ S \to S_2\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$.
Since $x \in \mathcal{L}(r)$, we have $x \in \mathcal{L}(r_1) \vee x \in \mathcal{L}(r_2)$.
By the strong inductive hypothesis, if $x \in \mathcal{L}(r_1)$, then it can be generated using grammar (V_1, Σ, S_1, P_1).
Let $S_1 \stackrel{*}{\Rightarrow} x$ be the sequence of productions that generates x in that grammar.
Then $S \Rightarrow S_1 \stackrel{*}{\Rightarrow} x$ is a valid sequence in grammar G.
A similar argument holds when $x \in \mathcal{L}(r_2)$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    for each $A \to \Lambda \in P_1$ **do**
        replace $A \to \Lambda$ by $A \to S_2$ in P_1
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
  $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
  $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
  for each $A \to \Lambda \in P_1$ **do**
    replace $A \to \Lambda$ by $A \to S_2$ in P_1
  return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
  $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
  $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
  for each $A \to \Lambda \in P_1$ **do**
    replace $A \to \Lambda$ by $A \to S_2$ in P_1
  return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.
By the strong inductive hypothesis, there is a sequence of production rules in P_1 such that $S_1 \stackrel{*}{\Rightarrow} x_1$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
$(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
for each $A \to \Lambda \in P_1$ **do**
replace $A \to \Lambda$ by $A \to S_2$ in P_1
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.
By the strong inductive hypothesis, there is a sequence of production rules in P_1 such that $S_1 \overset{*}{\Rightarrow} x_1$.
By Lemma 8.1, the sequence has the form $S_1 \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1$, for some $A \in V_1$, and $A \to \Lambda \in P_1$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
  $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
  $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
  for each $A \to \Lambda \in P_1$ **do**
    replace $A \to \Lambda$ by $A \to S_2$ in P_1
  return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.
By the strong inductive hypothesis, there is a sequence of production rules in P_1 such that $S_1 \overset{*}{\Rightarrow} x_1$.
By Lemma 8.1, the sequence has the form $S_1 \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1$, for some $A \in V_1$, and $A \to \Lambda \in P_1$.
By the strong inductive hypothesis, there is a sequence of production rules in P_2 such that $S_2 \overset{*}{\Rightarrow} x_2$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
$(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
$(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
for each $A \to \Lambda \in P_1$ **do**
replace $A \to \Lambda$ by $A \to S_2$ in P_1
return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.
By the strong inductive hypothesis, there is a sequence of production rules in P_1 such that $S_1 \overset{*}{\Rightarrow} x_1$.
By Lemma 8.1, the sequence has the form $S_1 \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1$, for some $A \in V_1$, and $A \to \Lambda \in P_1$.
By the strong inductive hypothesis, there is a sequence of production rules in P_2 such that $S_2 \overset{*}{\Rightarrow} x_2$.
By the construction of the new grammar, there is a rule $A \to S_2 \in P_1$.

Lemma 8.2

If $L = \mathcal{L}(r)$ for some Regular Expression r, then there exists a regular grammar G with unit productions such that $\mathcal{L}(G) = L$.

if $r = r_1 r_2$ **then**
    $(V_1, \Sigma, S_1, P_1) \Leftarrow$ Algorithm 8.1 (r_1)
    $(V_2, \Sigma, S_2, P_2) \Leftarrow$ Algorithm 8.1 (r_2)
    for each $A \to \Lambda \in P_1$ **do**
        replace $A \to \Lambda$ by $A \to S_2$ in P_1
    return $(V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1\})$

Claim: Let G be the returned grammar. If $x \in \mathcal{L}(r)$, then $x \in \mathcal{L}(G)$
Because $x \in \mathcal{L}(r)$, $\exists x_1, x_2$ such that $(x = x_1 x_2) \wedge (x_1 \in \mathcal{L}(r_1)) \wedge (x_2 \in \mathcal{L}(r_2))$.
By the strong inductive hypothesis, there is a sequence of production rules in P_1 such that $S_1 \overset{*}{\Rightarrow} x_1$.
By Lemma 8.1, the sequence has the form $S_1 \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1$, for some $A \in V_1$, and $A \to \Lambda \in P_1$.
By the strong inductive hypothesis, there is a sequence of production rules in P_2 such that $S_2 \overset{*}{\Rightarrow} x_2$.
By the construction of the new grammar, there is a rule $A \to S_2 \in P_1$.
We have the sequence: $S \Rightarrow S_1 \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1 S_2 \overset{*}{\Rightarrow} x_1 x_2$.