1. Grammars

2. Languages Generated by Grammars

3. Unsolvable Problems Concerning Grammars
Definition

A grammar is a semi-Thue process that involves two types of symbols:

1. **nonterminal symbols** or **variables** denoted by capital letters, $X, Y, Z, S, \ldots$, and
2. **terminal symbols** or **terminals** denoted by small letters, $a, b, c, \ldots$.

A special nonterminal symbol $S$ is the **start symbol**. In addition, for every production $x \rightarrow y$ the left part contains a nonterminal symbol.
A grammar will be denoted as

$$\Gamma = (\mathcal{V}, T, S, P),$$

where

- $\mathcal{V}$ is the set of non-terminals or variables;
- $T$ is the set of terminals;
- $S \in \mathcal{V}$ is the start symbol, and
- $P$ is the set of productions.
Definition

The language generated by $\Gamma$ is the set $L(\Gamma) \subseteq T^*$ given by

$$L(\Gamma) = \{ u \in T^* \mid S \overset{*}{\Rightarrow}^\Gamma u \}.$$ 

Note that in a grammar all non-terminal symbols are eliminated in the derivation process that ends up with a word over the terminal alphabet.
Example

Let \( \Gamma = (\{S, X, Y\}, \{a, b\}, S, \{S \rightarrow X, X \rightarrow aX, X \rightarrow 0, X \rightarrow Y, Y \rightarrow bY, Y \rightarrow 0\}) \).

Every derivation in \( \Gamma \) that begins with \( S \) and ends with a word in \( T^* \) has the form

\[
\begin{align*}
S & \Rightarrow_X \Gamma \Rightarrow aX \Rightarrow \Gamma aaX \\
& \Rightarrow \Gamma aaaX \Rightarrow \Gamma aaaY \Rightarrow \Gamma aaabY \\
& \Rightarrow \Gamma aaabbY \Rightarrow \Gamma aaabb.
\end{align*}
\]

Thus, the language \( L(\Gamma) \) is \( \{a^n b^m \mid n, m \in \mathbb{N}\} \).
Example

Let $\Gamma = (\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow 0\})$.
Every derivation in $\Gamma$ that begins with $S$ and ends with a word in $T^*$ has the form

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow \cdots \Rightarrow aaabbb.$$  

The language generated by this grammar is $L(\Gamma) = \{a^n b^n \mid n \in \mathbb{N}\}$. 
Example

Consider the grammar

$$\Gamma = (\{S, X, Y\}, \{a, b, c\}, S, P),$$

where $P$ consists of the following productions:

\[
\begin{align*}
\pi_0 : & \quad S \rightarrow abc, \\
\pi_1 : & \quad S \rightarrow aXbc, \\
\pi_2 : & \quad Xb \rightarrow bX, \\
\pi_3 : & \quad Xc \rightarrow Ybcc, \\
\pi_4 : & \quad bY \rightarrow Yb, \\
\pi_5 : & \quad aY \rightarrow aaX, \\
\pi_6 : & \quad aY \rightarrow aa
\end{align*}
\]

We will refer later in this lecture to this kind of grammars as \textbf{length-increasing grammars} because for each of its productions $x \rightarrow y$ we have $|x| \leq |y|$. 
We claim that \( L(\Gamma) = \{ a^n b^n c^n \mid n \in \mathbb{P} \} \).

- Any word \( \alpha \in \{ S, X, Y, a, b, c \}^* \) that occurs in a derivation, \( S \Rightarrow^* \alpha \) contains at most one nonterminal symbol.

- A derivation must end either by applying the production \( S \rightarrow abc \) or the production \( aY \rightarrow aa \) because only these productions allow us to eliminate a nonterminal symbol.

- If the last production is \( S \rightarrow abc \), then the derivation is \( S \Rightarrow abc \), and the derived word has the form prescribed. Otherwise, the symbol \( Y \) must be generated starting from \( S \), and the first production applied is \( S \rightarrow aXbc \).
Note that for every $i \geq 1$ we have

$$a^i X b^i c^i \Rightarrow^* \Gamma a^{i+1} X b^{i+1} c^{i+1}. $$

Indeed, we can write:

1. $a^i X b^i c^i \Rightarrow \pi_2 a^i b^i X c^i$
2. $a^i b^i X c^i \Rightarrow \pi_3 a^i b^i Ybc^{i+1}$
3. $a^i X b^i c^i \Rightarrow \pi_4 a^i Yb^{i+1} c^{i+1}$
4. $a^i Yb^{i+1} c^{i+1} \Rightarrow \pi_5 a^{i+1} X b^{i+1} c^{i+1}$

We claim that a word $\alpha$ contains the infix $aY$ (which allows us to apply the production $\pi_5$) and $S \Rightarrow^* \alpha$ if and only if $\alpha$ has the form $\alpha = a^i Yb^{i+1} c^{i+1}$ for some $i \geq 1$. 

An easy argument by induction on $i \geq 1$ allows us to show that if
\[ \alpha = a^i Y b^{i+1} c^{i+1} \]
then $S \xrightarrow{\ast} \Gamma \alpha$. We need to prove only the inverse implication. This can be done by strong induction on the length $n \geq 3$ of the derivation $S \xrightarrow{\ast} \Gamma \alpha$.

The shortest derivation that allows us to generate the word containing the infix $aY$ is

\[
S \xrightarrow{\Gamma} aXbc \xrightarrow{\Gamma} abXc \xrightarrow{\Gamma} abYbcc \xrightarrow{\Gamma} aYb^2c^2,
\]

and this word has the prescribed form.
Example cont’d

Suppose now that for derivations shorter than \( n \) the condition is satisfied, and let \( S \xRightarrow{\star} \alpha \) be a derivation of length \( n \) such that \( \alpha \) contains the infix \( aY \). By the inductive hypothesis the previous word in this derivation that contains the infix \( aY \) has the form \( \alpha' = a^jYb^{j+1}c^{j+1} \). To proceed from \( \alpha' \) we must apply the production \( \pi_5 \) and replace \( Y \) by \( X \). Thus, we have

\[
S \xRightarrow{\star} \alpha \xRightarrow{G} a^jYb^{j+1}c^{j+1} \xRightarrow{G} a^{j+1}Xb^{j+1}c^{j+1}.
\]
Example cont’d

Next, the symbol $X$ must “travel” to the right using the production $\pi_2$, transform itself into an $Y$ (when in touch with the $cs$) and $Y$ must “travel” to the left to create the infix $aY$. This can happen only through the application of the productions $\pi_3$ and $\pi_4$, as follows:

\[
\begin{align*}
\alpha_{j+1} & \Rightarrow \pi_2 \Rightarrow \beta_{j+1} \\
\beta_{j+1} & \Rightarrow \pi_3 \Rightarrow a^j b^j c^{j+1} \\
\beta_{j+1} & \Rightarrow \pi_4 \Rightarrow a^j b^j c^{j+2},
\end{align*}
\]

which proves that $\alpha$ has the desired form. Therefore, all the words in the language $L(\Gamma)$ have the form $a^n b^n c^n$. 

Theorem

Let $U$ be a language accepted by a nondeterministic Turing machine $M$. Then, there is a grammar $\Gamma$ such that $U = L(\Gamma)$. 
Proof

Recall that we defined a semi-Thue process $\Omega(\mathcal{M})$ attached to the TM $\mathcal{M}$. We started from $\mathcal{M}$ and defined first the semi-Thue system $\Sigma(\mathcal{M})$ on the alphabet

$$s_0, s_1, \ldots, s_K, q_0, q_1, \ldots, q_n, q_{n+1}, h$$

containing the following productions:

<table>
<thead>
<tr>
<th>Quadruple</th>
<th>Semi-Thue Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_i s_j s_k q_\ell$</td>
<td>$q_i s_j \rightarrow q_\ell s_k$</td>
</tr>
<tr>
<td>$q_i s_j q_\ell$</td>
<td>$q_i s_j s_k \rightarrow s_j q_\ell s_k, 0 \leq k \leq K$</td>
</tr>
<tr>
<td></td>
<td>$q_i s_j h \rightarrow s_j q_\ell s_0 h$</td>
</tr>
<tr>
<td>$q_i s_j L q_\ell$</td>
<td>$q_\ell s_k s_j \rightarrow s_0 q_\ell s_k, 0 \leq k \leq K$</td>
</tr>
<tr>
<td></td>
<td>$h q_i s_j \rightarrow h q_\ell s_0 s_j$</td>
</tr>
</tbody>
</table>
Proof cont’d

In addition we included in $\Sigma(\mathcal{M})$ the following productions:

- whenever $q_i s_j$ are not the first two symbols of a quadruple of $\mathcal{M}$ we place in $\Sigma(\mathcal{M})$ the production $q_i s_j \rightarrow q_{n+1} s_j$. Thus, $q_{n+1}$ serves as “halt” state.
- Finally, we place in $\Sigma(\mathcal{M})$ the productions:

$$q_{n+1} s_i \rightarrow q_{n+1}, 0 \leq i \leq K,$$
$$q_{n+1} h \rightarrow q_0 h,$$
$$s_i q_0 \rightarrow q_0, 0 \leq i \leq K.$$
Proof cont’d

The system $\Omega(M)$ contains the productions

<table>
<thead>
<tr>
<th>Quadruple</th>
<th>semi-Thue Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_i s_j s_k q_\ell$</td>
<td>$q_\ell s_k \rightarrow q_i s_j$</td>
</tr>
<tr>
<td>$q_i s_j R q_\ell$</td>
<td>$s_j q_\ell s_k \rightarrow q_i s_j s_k$, $0 \leq k \leq K$</td>
</tr>
<tr>
<td></td>
<td>$s_j q_\ell s_0 h \rightarrow q_i s_j h$</td>
</tr>
<tr>
<td>$q_i s_j L q_\ell$</td>
<td>$s_0 q_\ell s_k \rightarrow q_\ell s_k s_j$, $0 \leq k \leq K$</td>
</tr>
<tr>
<td></td>
<td>$h q_\ell s_0 s_j \rightarrow h q_i s_j$</td>
</tr>
</tbody>
</table>
In addition, we have in $\Omega(M)$:

- $q_{n+1}s_j \rightarrow q_is_j$ when $q_is_j$ are not the first two symbols of a quadruple of $M$, and

- $q_{n+1} \rightarrow q_{n+1}s_i, 0 \leq i \leq K,$
  $q_0h \rightarrow q_{n+1}h,$
  $q_0 \rightarrow s_iq_0, 0 \leq i \leq K.$
Proof cont’d

We construct the grammar $\Gamma$ by modifying the semi-Thue process $\Omega(M)$ as follows:

- the terminals of $\Gamma$ are just the letters of the alphabet $T = \{s_1, \ldots, s_m\}$ of $M$;
- the non-terminals (variables) of $\Gamma$ are the symbols of $\Omega(M)$ not in $T$, $s_0$, $q_0$, $q_n$, $q_{n+1}$, $h$;
- two additional non-terminals $S$ and $q$.

$S$ is the start symbol of $\Gamma$. 
Proof cont’d

The production of $\Gamma$ are:

- the productions of $\Omega(M)$;
- $S \rightarrow hq_0 h$;
- $hq_1 s_0 \rightarrow q$;
- $qs \rightarrow sq$ for each $s \in T$;
- $qh \rightarrow 0$. 
Suppose $M$ accepts $u \in T^*$, that is:

\[
S \xrightarrow[\Gamma]{} hq_0 h \xrightarrow[*][\Gamma]{} hq_1 s_0 u h \xrightarrow[\Gamma]{} q u h \xrightarrow[*][\Gamma]{} u q h \xrightarrow[\Gamma]{} u,
\]

so that $u \in L(\Gamma)$.  

Proof cont’d

Conversely, let $u \in L(\Gamma)$. Then, $u \in T^*$ and $S \xrightarrow{\Gamma}^* u$. Examining the list of productions of $\Gamma$ this derivation can be written as

$S \xrightarrow{\Gamma} hq_0 h \xrightarrow{\Gamma}^* vqhz \xrightarrow{\Gamma} vz = u.$

Note that $q$ could be introduced only by using the production $hq_1 s_0 \rightarrow q$. Thus, the derivation has the form

$S \xrightarrow{\Gamma} hq_0 h \xrightarrow{\Gamma}^* xhq_1 s_0 yhz \xrightarrow{\Gamma} xqyz \xrightarrow{\Gamma}^* xyqhz \xrightarrow{\Gamma} xyz = u,$

where $xy = v$. Thus, there is a derivation of $xhq_1 s_0 yhz$ from $hq_0 h$ in $\Gamma$. This derivation must actually be a derivation in $\Omega(M)$ because the added productions are inapplicable.
Proof cont’d

The productions in $\Omega(\mathcal{M})$ always lead from Post words to Post words, hence $xhq_1s_0yhz$ must be a Post word, which implies $x = z = 0$ and $u = xyz = y$. We conclude that

$$hq_0h \xrightarrow{\ast} hq_1s_0uh,$$

which implies that $\mathcal{M}$ accepts $u$. 
Let $\Gamma$ be a grammar having the alphabet

$$\{s_1, \ldots, s_n, V_1, \ldots, V_k\},$$

where $T = \{s_1, \ldots, s_n\}$ is the set of terminals and $\{V_1, \ldots, V_k\}$ is the set of variables (nonterminals). We assume that $S = V_1$ is the start symbol.

Assume that the alphabet of $\Gamma$ is ordered as above and we regard strings on this alphabet as integers in the base $n + k$. 
Theorem

The predicate \( u \Rightarrow_{\Gamma} v \) is primitive recursive.

Proof.

Let the production of \( \Gamma \) be \( x_i \rightarrow y_i \) for \( 1 \leq i \leq \ell \). For \( 1 \leq i \leq \ell \) define the predicate \( \text{PROD}_i(u, v) \) as

\[
(\exists r, s) \leq u [u = \text{CONCAT}(r, x_i, s) \& v = \text{CONCAT}(r, y_i, s)]
\]

Since \( \text{CONCAT} \) is primitive recursive, \( \text{PROD}_i \) is primitive recursive.

Since \( u \Rightarrow_{\Gamma} v \) if and only if

\[
\text{PROD}_1(u, v) \lor \text{PROD}_2(u, v) \lor \cdots \lor \text{PROD}_\ell(u, v)
\]

the result follows.
Define the predicate \( \text{DERIV}(u, y) \) to mean that for some \( m \) \( y = [u_1, \ldots, u_m, 1] \), where \( u_1, \ldots, u_m \) is a derivation of \( u \) from \( S \) in \( \Gamma \), that is,

\[
S = u_1 \Rightarrow \Gamma \quad \Rightarrow \quad u_2 \Rightarrow \Gamma \quad \cdots \quad \Rightarrow \quad u_m = u.
\]

1 has been added to avoid complications when \( u_m = u = 0 \). Note that the value of \( S \) in the base \( n + k \) is \( n + 1 \) (because \( S = V_1 \) is the \( n + 1 \)st in the alphabetic list).
Theorem

The predicate $\text{DERIV}(u, y)$ is primitive recursive.

Proof.

This follows from the following equivalent statements:

$$
\text{DERIV}(u, y) \iff \left( \exists m \right)_{\leq y} (m + 1 = Lt(y) \\
\& (y)_1 = n + 1 \& (y)_m = u \& (y)_{m+1} = 1 \\
\& (\forall j) < m (j = 0 \lor [(y)_j \Rightarrow (y)_{j+1}])
$$
Note that

- By the definition of $\text{DERIV}(u, y)$ we have

  $$S \xrightarrow{\Gamma}^* u \text{ if and only if } (\exists y)\text{DERIV}(u, y).$$

- $S \xrightarrow{\Gamma}^* u \text{ if and only if } \min_y \text{DERIV}(u, y) \downarrow.$

Therefore, $\{u \mid S \xrightarrow{\Gamma}^* u\}$ is recursively enumerable. Since

$L(\Gamma) = T^* \cap \{u \mid S \xrightarrow{\Gamma}^* u\}$ it follows that $L(\Gamma)$ is r.e.
Corollary

A language $U$ is r.e. if and only if there is a grammar $\Gamma$ such that $U = L(\Gamma)$. 
Putting together previous results we have the following

**Theorem**

*The following are equivalent for a language $L$:*

1. $L$ is r.e.;
2. $L$ is accepted by a deterministic TM;
3. $L$ is accepted by a nondeterministic TM;
4. there is a grammar $\Gamma$ such that $L = L(\Gamma)$. 
Definition

A grammar $\Gamma$ is called *length-increasing* if for every production $x \to y$ we have $|x| \leq |y|$.

An equivalent class of grammars to the class of length-increasing grammars is the class of *context-sensitive grammars*. This equivalence is a topic in the theory of formal languages.
Theorem

If $\Gamma$ is a length-increasing grammar, then the set
\[
\{ u \in (\mathcal{N} \cup \mathcal{T})^* \mid S \xrightarrow{\ast} \Gamma u \} \text{ is recursive.}
\]
Proof

Recall that we have shown that

\[ S \xrightarrow{\ast} u \text{ if and only if } \min_y \text{DERIV}(u, y) \downarrow \]

It will suffice to obtain a recursive bound for \( y \) to establish that \( L(\Gamma) \) is recursive.

Note that in every derivation in \( \Gamma \) we have

\[ 1 = |u_1| \leq |u_2| \leq \cdots \leq |u_m| = |u|. \]
Proof cont’d

Therefore, $u_1, u_2, \ldots, u_m = u \leq g(u)$, where $g(u)$ is the smallest number that represents a string of length $|u| + 1$ in the base $n + k$. Note that $g(u)$ is the value in the base $n + k$ of a string consisting of $|u| + 1$ repetitions of 1, so

$$g(u) = \sum_{i=0}^{\frac{|u|}{n+k}} (n + k)^i,$$

which is primitive recursive because $|u|$ is primitive recursive.

Note that we may assume that the derivation

$$S = u_1 \Rightarrow \Gamma u_2 \Rightarrow \Gamma \cdots \Rightarrow \Gamma u_m = u$$

contains no repetitions because given a sequence of steps

$$z = u_i \Rightarrow \Gamma u_{i+1} \Rightarrow \Gamma \cdots \Rightarrow \Gamma u_{i+\ell} = z$$

we could eliminate the steps $u_{i+1}, \ldots, u_\ell$. 
Thus, the length of the derivation is bounded by the total number of strings of length less or equal to $|u|$ on the alphabet with $n + k$ symbols, which is just the number $g(u)$. Hence,

$$[u_1, \ldots, u_m, 1] = \prod_{i=1}^{m} p_i^{u_i} \cdot p_{m+1} \leq h(u),$$

where

$$h(u) = \prod_{i=1}^{g(u)} p_i^{g(u)} \cdot p_{g(u)+1}. $$

Finally, we have $S \overset{*}{\Rightarrow} u$ if and only if $(\exists y)_{\leq h(u)} \text{DERIV}(u, y)$, which gives the result.
Theorem

If $\Gamma$ is a length-increasing grammar, then $L(\Gamma)$ is recursive.

Proof.

By the previous theorem, the set $\{u \in (V \cup T)^* \mid S \Rightarrow^* \Gamma u\}$ is recursive. Since

$$L(\Gamma) = \{u \in (V \cup T)^* \mid S \Rightarrow^* \Gamma u\} \cap T^*,$$

and $T^*$ is recursive, it follows that $L(\Gamma)$ is recursive. □
Let $M$ be a TM and let $u$ be a word in the alphabet of $M$. The grammar $\Gamma_u$ is constructed as follows:

- The variables of $\Gamma_u$ are the entire alphabet of $\Sigma(M)$ together with $S$ (the start symbol) and a new nonterminal symbol $V$. There is just one terminal symbol $a$.

- The production of $\Gamma_u$ are all productions of $\Sigma(M)$ together with

\[
S \rightarrow hq_1 s_0 uh, \quad hq_0 h \rightarrow V, \quad V \rightarrow aV, \quad V \rightarrow a
\]

We have $S \xrightarrow{\ast} \Gamma_u V$ if and only if $M$ accepts $u$. 
Lemma

If \( M \) accepts \( u \), then \( L(\Gamma_u) = \{ a^i \mid i \neq 0 \} \); if \( M \) does not accept \( u \), then \( L(\Gamma_u) = \emptyset \).

Proof.

The fact that \( M \) accepts \( u \) means that:

\[
S \xrightarrow{\ast} \Gamma_u hq_1 s_0 u h \xrightarrow{\Gamma_u} hq_0 h \xrightarrow{\Gamma_u} V \xrightarrow{\ast} \Gamma_u a^{n-1} V \xrightarrow{\Gamma_u} a^n,
\]

If \( M \) does not accept \( u \), then the word \( hq_0 u \) cannot be generated, so \( L(\Gamma_u) = \emptyset \).
Select $\mathcal{M}$ such that the language accepted by it is not recursive. Then, there is no algorithm for determining for given $u$ whether $\mathcal{M}$ accepts $u$. The lemma implies that

$$\mathcal{M} \text{ accepts } u \iff L(\Gamma_u) \neq \emptyset$$

$$\iff L(\Gamma_u) \text{ is infinite}$$

$$\iff a \in L(\Gamma_u).$$
The above prove the following:

**Theorem**

*There is no algorithm to determine of a given grammar $\Gamma$ whether*

1. $L(\Gamma)$ is empty;
2. $L(\Gamma)$ is infinite;
3. $v_0 \in L(\Gamma)$ for a fixed word $v_0$. 
There is no algorithm for determining of a given pair of grammars $\Gamma_1$ and $\Gamma_2$ whether

1. $L(\Gamma_1) \subseteq L(\Gamma_2)$;
2. $L(\Gamma_1) = L(\Gamma_2)$. 
Proof

Let $\Gamma_1$ be the grammar whose productions are

$$S \rightarrow aS, \ S \rightarrow a$$

We have $L(\Gamma_1) = \{a^i \mid i \neq 0\}$. Thus, by the previous theorem, $M$ accepts $u$ if and only if $L(\Gamma_1) = L(\Gamma_u)$ if and only if $L(\Gamma_1) \subseteq L(\Gamma_u)$. 