THEORY OF COMPUTATION
Grammars - 25

Prof. Dan A. Simovici

UMB

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 $x \to y$ 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 Γ is the set $L(\Gamma) \subseteq T^*$ given by

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

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 $\Gamma = (\{S, X, Y\}, \{a, b\}, S, \{S \to X, X \to aX, X \to 0, X \to Y, Y \to bY, Y \to 0\})$.
Every derivation in Γ that begins with S and ends with a word in T^* has the form

$$\begin{aligned} S &\underset{\Gamma}{\Rightarrow} X \underset{\Gamma}{\Rightarrow} aX \underset{\Gamma}{\Rightarrow} aaX \\ &\underset{\Gamma}{\Rightarrow} aaaX \underset{\Gamma}{\Rightarrow} aaaY \underset{\Gamma}{\Rightarrow} aaabY \\ &\underset{\Gamma}{\Rightarrow} aaabbY \underset{\Gamma}{\Rightarrow} aaabb. \end{aligned}$$

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

Example

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

$$S \underset{\Gamma}{\Rightarrow} aSb \underset{\Gamma}{\Rightarrow} aaSbb \underset{\Gamma}{\Rightarrow} aaaSbbb \underset{\Gamma}{\Rightarrow} aaabbb.$$

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:

$$\begin{array}{lcllcl}
\pi_0 & : & S \to abc, & \pi_1 & : & S \to aXbc, \\
\pi_2 & : & Xb \to bX, & \pi_3 & : & Xc \to Ybcc, \\
\pi_4 & : & bY \to Yb, & \pi_5 & : & aY \to aaX, \\
\pi_6 & : & aY \to aa & & &
\end{array}$$

We will refer later in this lecture to this kind of grammars as length-increasing grammars because for each of its productions $x \to y$ we have $|x| \leqslant |y|$.

Example cont'd

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, $S \overset{*}{\Rightarrow} \alpha$ contains at most one nonterminal symbol.
- A derivation must end either by applying the production $S \to abc$ or the production $aY \to aa$ because only these productions allow us to eliminate a nonterminal symbol.
- If the last production is $S \to abc$, 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 $S \to aXbc$.

Example cont'd

Note that for every $i \geqslant 1$ we have

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

Indeed, we can write:

$$\begin{array}{lllll} a^i X b^i c^i & \overset{i}{\underset{\pi_2}{\Rightarrow}} & a^i b^i X c^i & \overset{1}{\underset{\pi_3}{\Rightarrow}} & a^i b^i Y b c^{i+1} \\ & \overset{i}{\underset{\pi_4}{\Rightarrow}} & a^i Y b^{i+1} c^{i+1} & \overset{1}{\underset{\pi_5}{\Rightarrow}} & a^{i+1} X b^{i+1} c^{i+1} \end{array}$$

We claim that a word α contains the infix aY (which allows us to apply the production π_5) and $S \overset{*}{\underset{\Gamma}{\Rightarrow}} \alpha$ if and only if α has the form $\alpha = a^i Y b^{i+1} c^{i+1}$ for some $i \geqslant 1$.

Example cont'd

An easy argument by induction on $i \geqslant 1$ allows us to show that if $\alpha = a^i Y b^{i+1} c^{i+1}$ then $S \overset{*}{\underset{\Gamma}{\Rightarrow}} \alpha$. We need to prove only the inverse implication. This can be done by strong induction on the length $n \geqslant 3$ of the derivation $S \overset{*}{\underset{\Gamma}{\Rightarrow}} \alpha$.
The shortest derivation that allows us to generate the word containing the infix aY is

$$S \underset{\Gamma}{\Rightarrow} aXbc \underset{\Gamma}{\Rightarrow} abXc \underset{\Gamma}{\Rightarrow} abYbcc \underset{\Gamma}{\Rightarrow} aYb^2c^2,$$

and this word has the prescribed form.

Example cont'd

Suppose now that for derivations shorter than n the condition is satisfied, and let $S \overset{*}{\underset{G}{\Rightarrow}} \alpha$ be a derivation of length n such that α contains the infix aY. By the inductive hypothesis the previous word in this derivation that contains the infix aY has the form $\alpha' = a^j Y b^{j+1} c^{j+1}$. To proceed from α' we must apply the production π_5 and replace Y by X. Thus, we have

$$S \overset{*}{\underset{G}{\Rightarrow}} a^j Y b^{j+1} c^{j+1} \underset{G}{\Rightarrow} a^{j+1} X b^{j+1} c^{j+1}.$$

Example cont'd

Next, the symbol X must "travel" to the right using the production π_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 π_3 and π_4, as follows:

$$\begin{aligned} a^{j+1}Xb^{j+1}c^{j+1} & \overset{j+1}{\underset{\pi_2}{\Rightarrow}} a^{j+1}b^{j+1}Xc^{j+1} \\ & \overset{1}{\underset{\pi_3}{\Rightarrow}} a^{j+1}b^{j+1}Ybc^{j+2} \\ & \overset{j}{\underset{\pi_4}{\Rightarrow}} a^{j+1}Yb^{j+2}c^{j+2}, \end{aligned}$$

which proves that α 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 $\mathcal{M}$. Then, there is a grammar Γ 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:

Quadruple	semi-Thue Production
$q_i \ s_j \ s_k \ q_\ell$	$q_i s_j \to q_\ell s_k$
$q_i \ s_j \ R \ q_\ell$	$q_i s_j s_k \to s_j q_\ell s_k, 0 \leqslant k \leqslant K$
	$q_i s_j h \to s_j q_\ell s_0 h$
$q_i \ s_j \ L \ q_\ell$	$q_\ell s_k s_j \to s_0 q_\ell s_k, 0 \leqslant k \leqslant K$
	$h q_i s_j \to h q_\ell s_0 s_j$

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 $q_i s_j \to q_{n+1} s_j$. Thus, q_{n+1} serves as "halt" state.
- Finally, we place in $\Sigma(\mathcal{M})$ the productions:

$$\begin{aligned}
&q_{n+1} s_i \to q_{n+1}, 0 \leqslant i \leqslant K, \\
&q_{n+1} h \to q_0 h, \\
&s_i q_0 \to q_0, 0 \leqslant i \leqslant K.
\end{aligned}$$

Proof cont'd

The system $\Omega(\mathcal{M})$ contains the productions

Quadruple	semi-Thue Production
$q_i\ s_j\ s_k\ q_\ell$	$q_\ell s_k \to q_i s_j$
$q_i\ s_j\ R\ q_\ell$	$s_j q_\ell s_k \to q_i s_j s_k, 0 \leqslant k \leqslant K$
	$s_j q_\ell s_0 h \to q_i s_j h$
$q_i\ s_j\ L\ q_\ell$	$s_0 q_\ell s_k \to q_\ell s_k s_j, 0 \leqslant k \leqslant K$
	$h q_\ell s_0 s_j \to h q_i s_j$

Proof cont'd

In addition, we have in $\Omega(\mathcal{M})$:

- $q_{n+1}s_j \to q_i s_j$ when $q_i s_j$ are not the first two symbols of a quadruple of $\mathcal{M}$, and
- $$\begin{aligned} &q_{n+1} \to q_{n+1}s_i, 0 \leqslant i \leqslant K, \\ &q_0 h \to q_{n+1}h, \\ &q_0 \to s_i q_0, 0 \leqslant i \leqslant K. \end{aligned}$$

Proof cont'd

We construct the grammar Γ by modifying the semi-Thue process $\Omega(\mathcal{M})$ as follows:

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

S is the start symbol of Γ.

Proof cont'd

The production of Γ are:

- the productions of $\Omega(\mathcal{M})$;
- $S \to hq_0h$;
- $hq_1s_0 \to q$;
- $qs \to sq$ for each $s \in T$;
- $qh \to 0$.

Proof cont'd

Suppose $\mathcal{M}$ accepts $u \in T^*$, that is:

$$S \underset{\Gamma}{\Rightarrow} hq_0h \overset{*}{\underset{\Gamma}{\Rightarrow}} hq_1s_0uh \underset{\Gamma}{\Rightarrow} quh \overset{*}{\underset{\Gamma}{\Rightarrow}} uqh \underset{\Gamma}{\Rightarrow} u,$$

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

Proof cont'd

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

$$S \underset{\Gamma}{\Rightarrow} hq_0h \overset{*}{\underset{\Gamma}{\Rightarrow}} vqhz \underset{\Gamma}{\Rightarrow} vz = u.$$

Note that q could be introduced only by using the production $hq_1s_0 \to q$. Thus, the derivation has the form

$$S \underset{\Gamma}{\Rightarrow} hq_0h \overset{*}{\underset{\Gamma}{\Rightarrow}} xhq_1s_0yhz \underset{\Gamma}{\Rightarrow} xqyhz \overset{*}{\underset{\Gamma}{\Rightarrow}} xyqhz \underset{\Gamma}{\Rightarrow} xyz = u,$$

where $xy = v$. Thus, there is a derivation of xhq_1s_0yhz from hq_0h in Γ. This derivation must actually be a derivation in $\Omega(\mathcal{M})$ 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

$$hq_0h \overset{*}{\underset{\Omega(\mathcal{M})}{\Rightarrow}} hq_1s_0uh,$$

which implies that $\mathcal{M}$ accepts u.

Let Γ 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 Γ is ordered as above and we regard strings on this alphabet as integers in the base $n + k$.

Theorem

The predicate $u \underset{\Gamma}{\Rightarrow} v$ *is primitive recursive.*

Proof.

Let the production of Γ be $x_i \to y_i$ for $1 \leqslant i \leqslant \ell$. For $1 \leqslant i \leqslant \ell$ define the predicate $\mathrm{PROD}_i(u, v)$ as

$$(\exists r, s)_{\leqslant u}[u = \mathrm{CONCAT}(r, x_i, s) \& v = \mathrm{CONCAT}(r, y_i, s)]$$

Since CONCAT is primitive recursive, PROD_i is primitive recursive. Since $u \underset{\Gamma}{\Rightarrow} v$ if and only if

$$\mathrm{PROD}_1(u, v) \vee \mathrm{PROD}_2(u, v) \vee \cdots \vee \mathrm{PROD}_\ell(u, v)$$

the result follows. $\square$

Define the predicate 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 Γ, that is,

$$S = u_1 \underset{\Gamma}{\Rightarrow} u_2 \underset{\Gamma}{\Rightarrow} \cdots \underset{\Gamma}{\Rightarrow} u_m = u.$$

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 $(n+1)^{\mathrm{st}}$ symbol in the alphabetic list).

Theorem

The predicate DERIV(u, y) is primitive recursive.

Proof.

This follows from the following equivalent statements:

$$
\begin{array}{rcl}
\mathrm{DERIV}(u, y) & \Leftrightarrow & (\exists m)_{\leqslant y}(m + 1 = Lt(y) \\
 & \& & (y)_1 = n + 1 \& (y)_m = u \& (y)_{m+1} = 1 \\
 & \& & (\forall j)_{<m}(j = 0 \vee [(y)_j \underset{\Gamma}{\Rightarrow} (y)_{j+1}])
\end{array}
$$

$\square$

Note that

- By the definition of DERIV(u, y) we have

$$S \overset{*}{\underset{\Gamma}{\Rightarrow}} u \text{ if and only if } (\exists y)\text{DERIV}(u, y).$$

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

Therefore, $\{u \mid S \overset{*}{\underset{\Gamma}{\Rightarrow}} u\}$ is recursively enumerable. Since $L(\Gamma) = T^* \cap \{u \mid S \overset{*}{\underset{\Gamma}{\Rightarrow}} u\}$ it follows that $L(\Gamma)$ is r.e.

Corollary

A language U is r.e. if and only if there is a grammar Γ 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 Γ such that $L = L(\Gamma)$.*

Definition

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

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

Theorem

If Γ is a length-increasing grammar, then the set $\{u \in (\mathcal{V} \cup T)^* \mid S \overset{*}{\underset{\Gamma}{\Rightarrow}} u\}$ is recursive.

Proof

Recall that we have shown that

$$S \overset{*}{\underset{\Gamma}{\Rightarrow}} u \text{ if and only if } \min_{y} \mathrm{DERIV}(u, y) \downarrow$$

It will suffice to obtain a recursive bound for y to establish that $L(\Gamma)$ is recursive.
Note that in every derivation in Γ we have

$$1 = |u_1| \leqslant |u_2| \leqslant \cdots \leqslant |u_m| = |u|.$$

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

Proof cont'd

Note that:

- $g(u)$ is the value in the base $n+k$ of a string consisting of $|u|+1$ repetitions of 1, so $g(u) = \sum_{i=0}^{|u|}(n+k)^i$, which is primitive recursive because $|u|$ is primitive recursive.
- We may assume that the derivation

$$S = u_1 \underset{\Gamma}{\Rightarrow} u_2 \underset{\Gamma}{\Rightarrow} \cdots \underset{\Gamma}{\Rightarrow} u_m = u$$

contains no repetitions because given a sequence of steps

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

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,

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

where

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

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

Theorem

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

Proof.

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

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

and T^* is recursive, it follows that $L(\Gamma)$ is recursive. $\square$

Let $\mathcal{M}$ be a TM and let u be a word in the alphabet of $\mathcal{M}$. The grammar Γ_u is constructed as follows:

- The variables of Γ_u are the entire alphabet of $\Sigma(\mathcal{M})$ together with S (the start symbol) and a new nonterminal symbol V. There is just one terminal symbol a.
- The production of Γ_u are all productions of $\Sigma(\mathcal{M})$ together with

$$S \to hq_1 s_0 uh, hq_0 h \to V, V \to aV, V \to a$$

We have $S \overset{*}{\underset{\Gamma_u}{\Rightarrow}} V$ if and only if $\mathcal{M}$ accepts u.

Lemma

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

Proof.

The fact that $\mathcal{M}$ accepts u means that:

$$S \overset{*}{\underset{\Gamma_u}{\Rightarrow}} hq_1 s_0 uh \underset{\Gamma_u}{\Rightarrow} hq_0 h \underset{\Gamma_u}{\Rightarrow} V \overset{*}{\underset{\Gamma_u}{\Rightarrow}} a^{n-1}V \underset{\Gamma_u}{\Rightarrow} a^n,$$

If $\mathcal{M}$ 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

$$\begin{aligned} \mathcal{M} \text{ accepts } u \quad &\Leftrightarrow \quad L(\Gamma_u) \neq \emptyset \\ &\Leftrightarrow \quad L(\Gamma_u) \text{ is infinite} \\ &\Leftrightarrow \quad a \in L(\Gamma_u). \end{aligned}$$

The above prove the following:

Theorem

There is no algorithm to determine of a given grammar Γ *whether*

1. $L(\Gamma)$ *is empty;*
2. $L(\Gamma)$ *is infinite;*
3. $v_0 \in L(\Gamma)$ *for a fixed word* v_0.

Theorem

There is no algorithm for determining of a given pair of grammars Γ_1 and Γ_2 whether

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

Proof

Let Γ_1 be the grammar whose productions are

$$S \to aS, S \to a$$

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