Production Grammars

FOCS, Fall 2020

Computation as string rewriting

Turing machines and machine models in general are based on the notion of "recognizing" a string by running a machine

This week - a new model of computation based on string generation via rewriting

We start with a simpler model than the full rewriting model

- context-free grammars
- then unrestricted grammars

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow a\ S\ b$

$S \rightarrow \varepsilon$

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

input alphabet (terminal symbols)

Start symbol = S

$S \rightarrow a\, S\, b$

$S \rightarrow \varepsilon$

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

Start symbol = S

Production rules : $a \rightarrow u$

$S \rightarrow a\,S\,b$

$S \rightarrow \varepsilon$

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

Start symbol = S

Nonterminal symbols

$S \rightarrow a\ S\ b$

$S \rightarrow \varepsilon$

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

Start symbol = S

Dedicated nonterminal symbols to start the process

$S \rightarrow a\,S\,b$

$S \rightarrow \varepsilon$

Example 1: $\{ a^n b^n \mid n \geq 0 \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow a\,S\,b$

$S \rightarrow \varepsilon$

Process:

- put down the start symbol as the current string
- replace any of the nonterminals in the string using any of the rules that applies
- stop when you have a string with only terminal symbols

Any string you can generate by this process is said to be generated by the grammar

Example 1: generation

The string aaabbb is generated by the previous grammar:

S

Example 1: generation

The string aaabbb is generated by the previous grammar:

$S \quad \Rightarrow a\ S\ b$

Example 1: generation

The string aaabbb is generated by the previous grammar:

$$
\begin{aligned}
S &\Rightarrow a\ S\ b \\
  &\Rightarrow a\ a\ S\ b\ b
\end{aligned}
$$

Example 1: generation

The string aaabbb is generated by the previous grammar:

$$\begin{aligned} S &\Rightarrow a\ S\ b \\ &\Rightarrow a\ a\ S\ b\ b \\ &\Rightarrow a\ a\ a\ S\ b\ b\ b \end{aligned}$$

Example 1: generation

The string aaabbb is generated by the previous grammar:

$$
\begin{aligned}
S &\Rightarrow a\ S\ b \\
  &\Rightarrow a\ a\ S\ b\ b \\
  &\Rightarrow a\ a\ a\ S\ b\ b\ b \\
  &\Rightarrow a\ a\ a\ b\ b\ b
\end{aligned}
$$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \to T\,U$

$T \to a\,T\,b$

$T \to \varepsilon$

$U \to b\,U$

$U \to \varepsilon$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

S

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\,U$

$T \rightarrow a\,T\,b$

$T \rightarrow \varepsilon$

$U \rightarrow b\,U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$S \Rightarrow T\,U$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$\begin{aligned} S &\Rightarrow T\ U \\ &\Rightarrow a\ T\ b\ U \end{aligned}$$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$\begin{aligned} S &\Rightarrow T\ U \\ &\Rightarrow a\ T\ b\ U \\ &\Rightarrow a\ a\ T\ b\ b\ U \end{aligned}$$

Example 2: { $a^n b^m$ | m ≥ n }

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$\begin{aligned} S &\Rightarrow T\ U \\ &\Rightarrow a\ T\ b\ U \\ &\Rightarrow a\ a\ T\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ U \end{aligned}$$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$
\begin{aligned}
S &\Rightarrow T\ U \\
&\Rightarrow a\ T\ b\ U \\
&\Rightarrow a\ a\ T\ b\ b\ U \\
&\Rightarrow a\ a\ b\ b\ U \\
&\Rightarrow a\ a\ b\ b\ b\ U
\end{aligned}
$$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$

$T \rightarrow a\ T\ b$

$T \rightarrow \varepsilon$

$U \rightarrow b\ U$

$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$\begin{aligned} S &\Rightarrow T\ U \\ &\Rightarrow a\ T\ b\ U \\ &\Rightarrow a\ a\ T\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ b\ b\ U \end{aligned}$$

Example 2: $\{ a^n b^m \mid m \geq n \}$

$\Sigma = \{ a, b \}$

Start symbol = S

$S \rightarrow T\ U$
$T \rightarrow a\ T\ b$
$T \rightarrow \varepsilon$
$U \rightarrow b\ U$
$U \rightarrow \varepsilon$

aabbbb generated by grammar:

$$\begin{aligned} S &\Rightarrow T\ U \\ &\Rightarrow a\ T\ b\ U \\ &\Rightarrow a\ a\ T\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ b\ b\ U \\ &\Rightarrow a\ a\ b\ b\ b\ b \end{aligned}$$

Example 3: { w | w = rev(w) }

Σ = { a, b }

Start symbol = S

$S \to a\ S\ a$

$S \to b\ S\ b$

$S \to a$

$S \to b$

$S \to \varepsilon$

Example 3: { w | w = rev(w) }

$\Sigma = \{ a, b \}$

Start symbol = S

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

$S \rightarrow b\,S\,b$

$S \rightarrow a$

$S \rightarrow b$

$S \rightarrow \varepsilon$

abaaaba generated by grammar:

$$\begin{aligned} S &\Rightarrow a\,S\,a \\ &\Rightarrow a\,b\,S\,b\,a \\ &\Rightarrow a\,b\,a\,S\,a\,b\,a \\ &\Rightarrow a\,b\,a\,a\,a\,b\,a \end{aligned}$$

Context-free grammars

Definition: A context-free production grammar is a tuple $G = (N, \Sigma, R, S)$ where

N is a finite set of nonterminal symbols

Σ is a finite set of terminal symbols (input alphabet)

R is a finite set of rules of the form $a \rightarrow w$ (where $a \in N$, $w \in (N \cup \Sigma)^*$)

S is the start symbol ($S \in N$)

Definition of rewriting

One-step rewrite relation $u \Rightarrow v$

$u_1\ a\ v_1 \Rightarrow u_1\ w\ v_1 \quad \text{if } a \rightarrow w \in R$

Multi-step rewrite relation $u \Rightarrow^* v$

$u \Rightarrow^* v$ if $u = v$ or $u \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_k \Rightarrow v$ for some $w_1, \ldots, w_k$ $(k > 0)$

Context-free language

Language of G:

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$

A language A is context-free if there is a context-free grammar that can generate exactly the strings in A

Every regular language is context-free

- can transform every finite state machine M into a context-free grammar that generate the same language accepted by M

Simulating finite automata using context-free grammars

Consider a finite state machine $M = (Q, \Sigma, \Delta, s, F)$

Associate with the states $\{p, q, r, ...\}$ of Q some nonterminals N_p , N_q , N_r, …

Consider the grammar G_M with rules:

$N_p \rightarrow a\ N_q$	for every $(p, a, q) \in \Delta$
$N_p \rightarrow \varepsilon$	for every $p \in F$

Let N_s be the start symbol

Simulating finite automata using context-free grammars

Consider a finite state machine $M = (Q, \Sigma, \Delta, s, F)$

Associate with the states $\{p, q, r, ...\}$ of Q some nonterminals N_p , N_q , N_r, …

Consider the grammar G_M with rules:

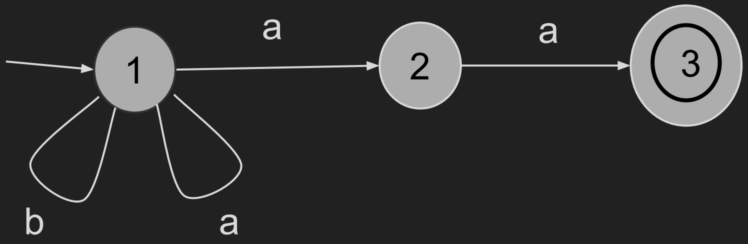
$N_p \to a\ N_q$	for every $(p, a, q) \in \Delta$
$N_p \to \varepsilon$	for every $p \in F$

Let N_s be the start symbol

Claim: G_M generates the language accepted by M

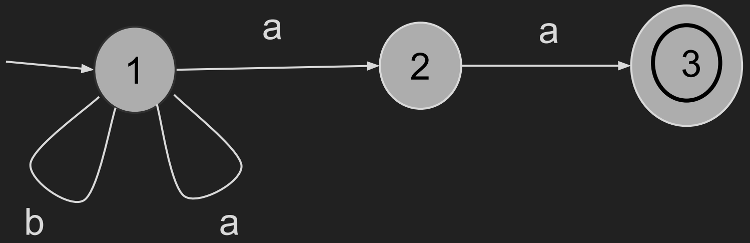
Example

$\Sigma = \{ a, b \}$



Example

$\Sigma = \{ a, b \}$



$N_1 \to b\ N_1$

$N_1 \to a\ N_1$

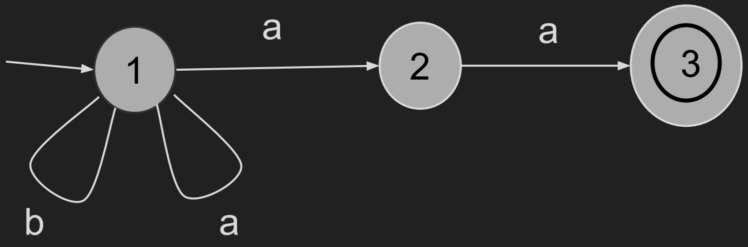
$N_1 \to a\ N_2$

$N_2 \to a\ N_3$

$N_3 \to \varepsilon$

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\, N_1$

$N_1 \rightarrow a\, N_1$

$N_1 \rightarrow a\, N_2$

$N_2 \rightarrow a\, N_3$

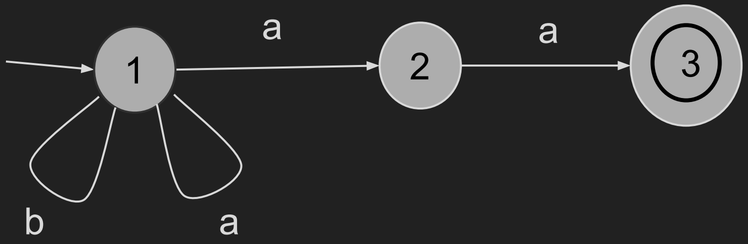
$N_3 \rightarrow \varepsilon$

N_1

The grammar generates ababaa:

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\,N_1$

$N_1 \rightarrow a\,N_1$

$N_1 \rightarrow a\,N_2$

$N_2 \rightarrow a\,N_3$

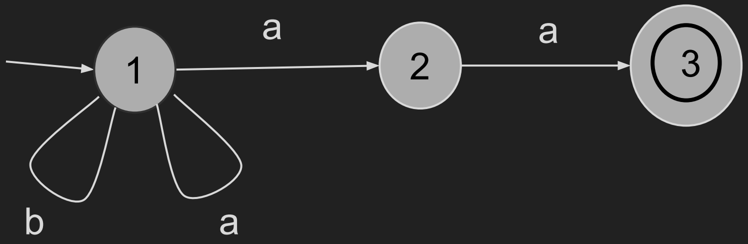
$N_3 \rightarrow \varepsilon$

$N_1 \Rightarrow a\,N_1$

The grammar generates ababaa:

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\ N_1$

$N_1 \rightarrow a\ N_1$

$N_1 \rightarrow a\ N_2$

$N_2 \rightarrow a\ N_3$

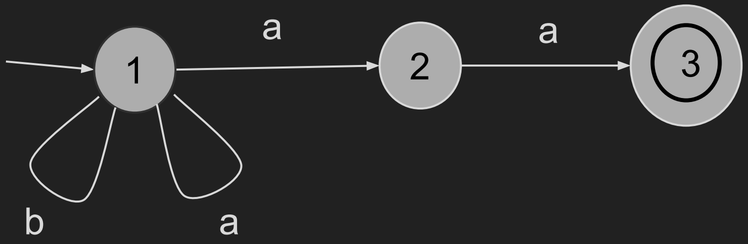
$N_3 \rightarrow \varepsilon$

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \end{aligned}$$

The grammar generates ababaa:

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\ N_1$

$N_1 \rightarrow a\ N_1$

$N_1 \rightarrow a\ N_2$

$N_2 \rightarrow a\ N_3$

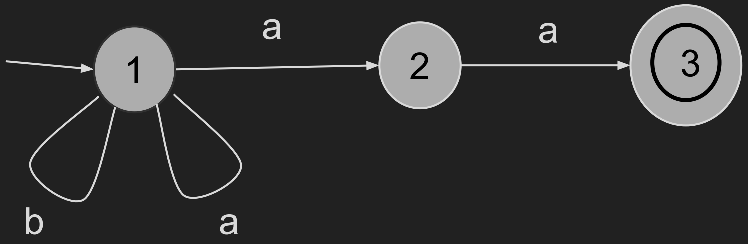
$N_3 \rightarrow \varepsilon$

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ N_1 \end{aligned}$$

The grammar generates ababaa:

Example

$\Sigma = \{ a, b \}$



$N_1 \to b\ N_1$

$N_1 \to a\ N_1$

$N_1 \to a\ N_2$

$N_2 \to a\ N_3$

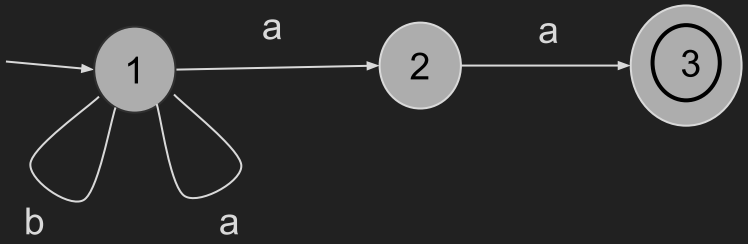
$N_3 \to \varepsilon$

The grammar generates ababaa:

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ N_1 \\ &\Rightarrow a\ b\ a\ b\ N_1 \end{aligned}$$

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\ N_1$

$N_1 \rightarrow a\ N_1$

$N_1 \rightarrow a\ N_2$

$N_2 \rightarrow a\ N_3$

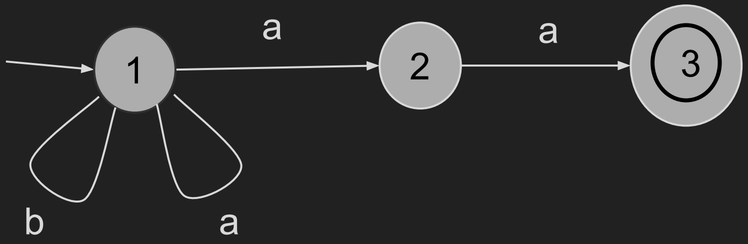
$N_3 \rightarrow \varepsilon$

The grammar generates ababaa:

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ N_1 \\ &\Rightarrow a\ b\ a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ b\ a\ N_2 \end{aligned}$$

Example

$\Sigma = \{ a, b \}$



$N_1 \to b\ N_1$

$N_1 \to a\ N_1$

$N_1 \to a\ N_2$

$N_2 \to a\ N_3$

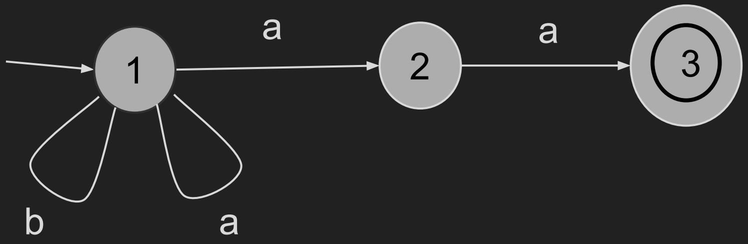
$N_3 \to \varepsilon$

The grammar generates ababaa:

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ N_1 \\ &\Rightarrow a\ b\ a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ b\ a\ N_2 \\ &\Rightarrow a\ b\ a\ b\ a\ a\ N_3 \end{aligned}$$

Example

$\Sigma = \{ a, b \}$



$N_1 \rightarrow b\ N_1$

$N_1 \rightarrow a\ N_1$

$N_1 \rightarrow a\ N_2$

$N_2 \rightarrow a\ N_3$

$N_3 \rightarrow \varepsilon$

The grammar generates ababaa:

$$\begin{aligned} N_1 &\Rightarrow a\ N_1 \\ &\Rightarrow a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ N_1 \\ &\Rightarrow a\ b\ a\ b\ N_1 \\ &\Rightarrow a\ b\ a\ b\ a\ N_2 \\ &\Rightarrow a\ b\ a\ b\ a\ a\ N_3 \\ &\Rightarrow a\ b\ a\ b\ a\ a \end{aligned}$$

Unrestricted production grammars

Context-free production grammars are nice and simple, but they cannot generate all computable languages

For instance, they cannot generate $\{ a^n b^n c^n \mid n \geq 0 \}$ which is certainly Turing-computable

We can extend production grammars to generate all computable languages:

- allow productions to take the form $u \to w$
  where $u, w \in (N \cup \Sigma)^*$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

grammar generates aaabbbccc:

S

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$S \Rightarrow T\ W$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$\begin{aligned} S &\Rightarrow T\ W \\ &\Rightarrow a\ T\ b\ X\ W \end{aligned}$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$\begin{aligned} S &\Rightarrow T\ W \\ &\Rightarrow a\ T\ b\ X\ W \\ &\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \end{aligned}$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$\begin{aligned} S &\Rightarrow T\ W \\ &\Rightarrow a\ T\ b\ X\ W \\ &\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\ &\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \end{aligned}$$

Example: { $a^nb^nc^n$ | $n \geq 0$ }

Σ = { a, b, c }

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$\begin{aligned} S &\Rightarrow T\ W \\ &\Rightarrow a\ T\ b\ X\ W \\ &\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\ &\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\ &\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \end{aligned}$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c
\end{aligned}
$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ X\ W\ c
\end{aligned}
$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ X\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ W\ c\ c
\end{aligned}
$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$
$T \rightarrow a\ T\ b\ X$
$T \rightarrow \varepsilon$
$X\ b \rightarrow b\ X$
$X\ W \rightarrow W\ c$
$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ X\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ X\ b\ W\ c\ c
\end{aligned}$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ X\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ X\ b\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ b\ X\ W\ c\ c
\end{aligned}
$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\, W$

$T \rightarrow a\, T\, b\, X$

$T \rightarrow \varepsilon$

$X\, b \rightarrow b\, X$

$X\, W \rightarrow W\, c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\,W \\
&\Rightarrow a\,T\,b\,X\,W \\
&\Rightarrow a\,a\,T\,b\,X\,b\,X\,W \\
&\Rightarrow a\,a\,a\,T\,b\,X\,b\,X\,b\,X\,W \\
&\Rightarrow a\,a\,a\,b\,X\,b\,X\,b\,X\,W \\
&\Rightarrow a\,a\,a\,b\,X\,b\,X\,b\,W\,c \\
&\Rightarrow a\,a\,a\,b\,X\,b\,b\,X\,W\,c \\
&\Rightarrow a\,a\,a\,b\,X\,b\,b\,W\,c\,c \\
&\Rightarrow a\,a\,a\,b\,b\,X\,b\,W\,c\,c \\
&\Rightarrow a\,a\,a\,b\,b\,b\,X\,W\,c\,c \\
&\Rightarrow a\,a\,a\,b\,b\,b\,W\,c\,c\,c
\end{aligned}
$$

Example: $\{ a^n b^n c^n \mid n \geq 0 \}$

$\Sigma = \{ a, b, c \}$

Starting symbol = S

$S \rightarrow T\ W$

$T \rightarrow a\ T\ b\ X$

$T \rightarrow \varepsilon$

$X\ b \rightarrow b\ X$

$X\ W \rightarrow W\ c$

$W \rightarrow \varepsilon$

grammar generates aaabbbccc:

$$
\begin{aligned}
S &\Rightarrow T\ W \\
&\Rightarrow a\ T\ b\ X\ W \\
&\Rightarrow a\ a\ T\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ T\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ X\ W \\
&\Rightarrow a\ a\ a\ b\ X\ b\ X\ b\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ X\ W\ c \\
&\Rightarrow a\ a\ a\ b\ X\ b\ b\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ X\ b\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ b\ X\ W\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ b\ W\ c\ c\ c \\
&\Rightarrow a\ a\ a\ b\ b\ b\ c\ c\ c
\end{aligned}
$$

Production grammar

Definition: An (unrestricted) production grammar is a tuple $G = (N, \Sigma, R, S)$ where

- N is a finite set of nonterminal symbols
- Σ is a finite set of terminal symbols (input alphabet)
- R is a finite set of rules of the form $u \rightarrow w$ (where $u, w \in (N \cup \Sigma)^*$)
- S is the start symbol ($S \in N$)

Essentially same definition of rewriting $\Rightarrow^*$ and $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$

Production grammars and Turing machines

You can write a Turing machine that accepts exactly the strings generated by the production grammar

- 3-tape Turing machine
- copy the input string on the third tape
- apply rewrites in breadth-first order using second tape as a queue
- after each rewrite check if you have the same string as on the third tape

Not hard, really but annoying to write

Also not surprising: by Church-Turing, we should be able to implement grammars

Simulating a Turing machine with a production grammar

Given a Turing machine M, derive a grammar G_M that generates any string that M can accept by essentially using rewrite rules to rewrite a string in a way that mimics how the Turing machines "rewrites" its tape

Shares some ideas with the proof of uncomputability of PCP

Slight trickiness: we need the grammar to use the string as the "tape" of the Turing machine, while still in the end generating the input string that we started simulating the Turing machine with

The grammar corresponding to a Turing machine

Given $M = (Q, \Gamma, \Sigma, \delta, q_s, q_{acc}, q_{rej})$

Construct $G_M = (N, \Sigma, R, A)$

where nonterminals in N are either:

- one of A, B, C
- a state in Q
- a pair $[a, x]$ where $a \in \Sigma \cup \{\varepsilon\}$ and $x \in \Gamma$

A is the start symbol
and production rules are given on the next few slides

Rules to set up the tape

$A \rightarrow q_s\ [\varepsilon, >]\ B$

$B \rightarrow [a, a]\ B$ one rule for every $a \in \Sigma$

$B \rightarrow C$

$C \rightarrow [\varepsilon, \_]\ C$

$C \rightarrow \varepsilon$

Rules to simulate the Turing machine

q [a, x] $\rightarrow$ [a, y] p

one rule for every ∂(q, x) = (p, y, R) and every a

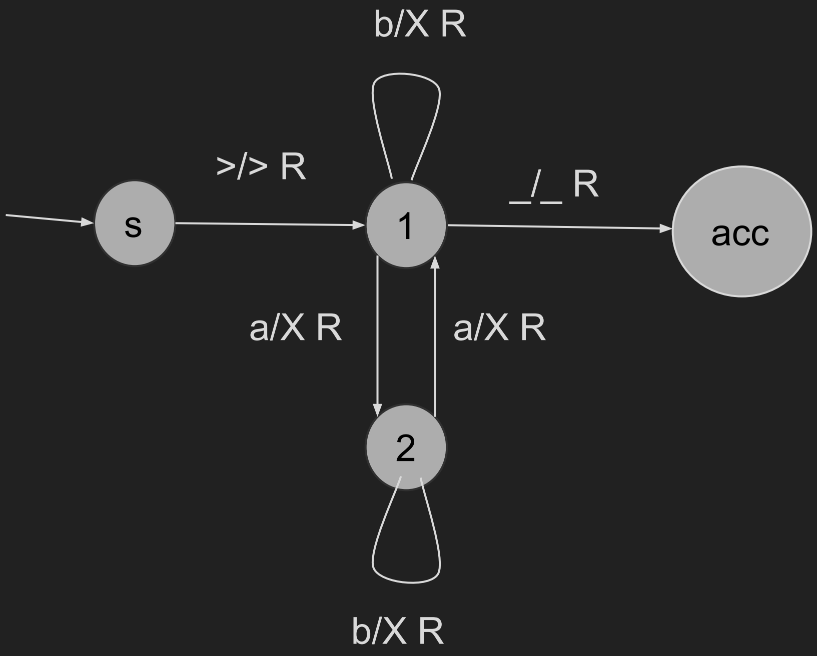
[b, z] q [a, x] $\rightarrow$ p [b, z] [a, y]

one rule for every ∂(q, x) = (p, y, L) and every a, b, z

Rules to clean up at the end

$[\varepsilon, x]\ q_{acc} \rightarrow q_{acc}$	one rule for every x
$q_{acc}\ [\varepsilon, x] \rightarrow q_{acc}$	one rule for every x
$[a, x]\ q_{acc} \rightarrow q_{acc}\ a\ q_{acc}$	one rule for every a, x
$q_{acc}\ [a, x] \rightarrow q_{acc}\ a\ q_{acc}$	one rule for every a, x
$q_{acc} \rightarrow \varepsilon$	

Example

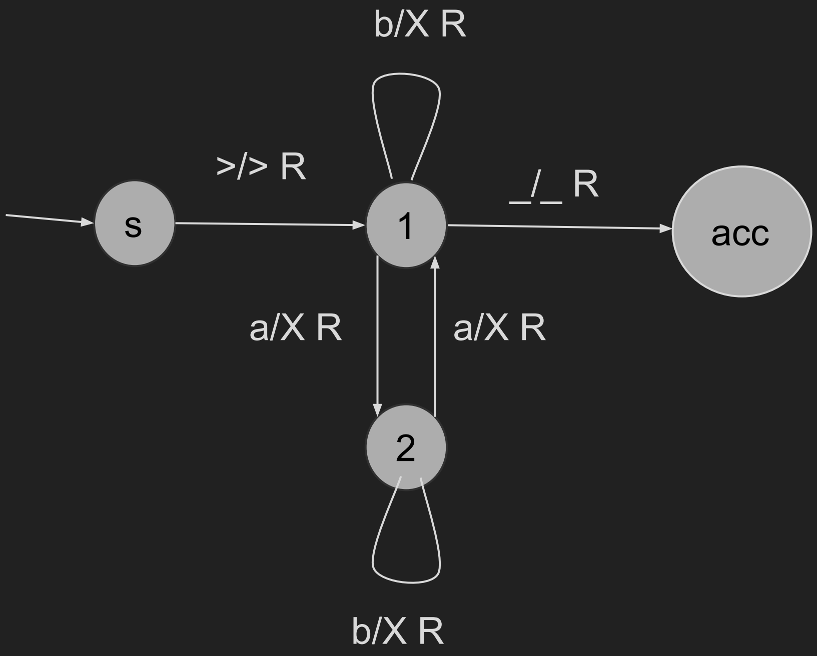


Σ = { a, b }

Γ = { a, b, X, >, _ }

N = { A, B, C,
s, 1, 2, acc,
[ε, a], [ε, b], [ε, X] [ε, >], [ε, _],
[a, a], [a, b], [a, X], [a, >], [a, _],
[b, a], [b, b], [b, X], [b, >], [b, _] }

Example



A ⟶ s [ε, >] B
B ⟶ [a, a] B
B ⟶ [b, b] B
B ⟶ C
C ⟶ [ε, _] C
C ⟶ ε

s [ε. >] ⟶ [ε, >] 1
1 [a, a] ⟶ [a, X] 2
1 [b, a] ⟶ [b, X] 2
1 [a, b] ⟶ [a, X] 1
1 [b, b] ⟶ [b, X] 1
2 [a, a] ⟶ [a. X] 1
2 [b, a] ⟶ [b, X] 1
2 [a, b] ⟶ [a, X] 2
2 [b, b] ⟶ [b, X] 2
1 [a, _] ⟶ [a, _] acc
1 [b, _] ⟶ [b, _] acc
1 [ε, _] ⟶ [ε, _] acc

[ε, >] acc ⟶ acc
[ε, _] acc ⟶ acc
[ε, a] acc ⟶ acc
[ε, b] acc⟶ acc
[ε, X] acc ⟶ acc
[a, >] acc ⟶ acc a acc
[b, >] acc ⟶ acc b acc
[a, _] acc ⟶ acc a acc
[b, _] acc ⟶ acc b acc
[a, a] acc ⟶ acc a acc
[b, a] acc ⟶ acc b acc
[a, b] acc ⟶ acc a acc
[b, b] acc ⟶ acc b acc
[a, X] acc ⟶ acc a acc
[b, X] acc ⟶ acc b acc
acc ⟶ ε

… (unused rules skipped)

The Turing machine accepts abaaa

A ⇒ s [ε, >] B
⇒ s [ε, >] [a, a] B
⇒ s [ε, >] [a, a] [b, b] B
⇒ s [ε, >] [a, a] [b, b] [a, a] B
⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] B
⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] [a, a] B
⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] [a, a] C
⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] [a, a] [ε, _] C
⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] [a, a] [ε, _]

The Turing machine accepts abaaa

...

⇒ s [ε, >] [a, a] [b, b] [a, a] [a, a] [a, a] [ε, _]

⇒ [ε, >] 1 [a, a] [b, b] [a, a] [a, a] [a, a] [ε, _]

⇒ [ε, >] [a, X] 2 [b, b] [a, a] [a, a] [a, a] [ε, _]

⇒ [ε, >] [a, X] [b, X] 2 [a, a] [a, a] [a, a] [ε, _]

⇒ [ε, >] [a, X] [b, X] [a, X] 1 [a, a] [a, a] [ε, _]

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] 2 [a, a] [ε, _]

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] [a, X] 1 [ε, _]

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] [a, X] [ε, _] acc

The Turing machine accepts abaaa

…

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] [a, X] [ε, _] acc

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] [a, X] acc

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] acc a acc

⇒ [ε, >] [a, X] [b, X] [a, X] [a, X] acc a

⇒ [ε, >] [a, X] [b, X] [a, X] acc a acc a

⇒ [ε, >] [a, X] [b, X] [a, X] acc a a

⇒ [ε, >] [a, X] [b, X] acc a acc a a

⇒ [ε, >] [a, X] [b, X] acc a a a

⇒ [ε, >] [a, X] acc b acc a a a

The Turing machine accepts abaaa

…

⇒ [ε, >] [a, X] acc b acc a a a

⇒ [ε, >] [a, X] acc b a a a

⇒ [ε, >] acc a acc b a a a

⇒ [ε, >] acc a b a a a

⇒ acc a b a a a

⇒ a b a a a

Conclusion

The problem can grammar G generate string w? is uncomputable

- if it were, we could solve the problem MP = { <M, w> | M accepts w }
- given <M, w>, we could answer "does M accept w" by creating G_M and asking "can G_M generate string w"?
  - if yes, then M accepts w
  - if no, then M does not accept w
- this would let us compute MP, which we know is uncomputable

Note: If we restrict to *context-free production grammars*, the question can context-free grammar G generate string w? becomes computable