Nonregular languages

Nonregular languages

Theorem (Pumping Lemma)

Nonregular languages

every long enough word of a
regular language can be pumped

Theorem (Pumping Lemma)

Nonregular languages

every long enough word of a regular language can be pumped

Theorem (Pumping Lemma)

If L is a regular language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and

1. $xy^iz \in L$, for all $i \in \mathbb{N}$
2. $|y| > 0$
3. $|xy| \leq p$

Nonregular languages

every long enough word of a regular language can be pumped

Theorem (Pumping Lemma)

If L is a regular language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and

1. $xy^iz \in L$, for all $i \in \mathbb{N}$
2. $|y| > 0$
3. $|xy| \leq p$

Proof sketch easy, using the pigeonhole principle

Nonregular languages

every long enough word of a regular language can be pumped

Theorem (Pumping Lemma)

If L is a regular language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and

1. $xy^iz \in L$, for all $i \in \mathbb{N}$
2. $|y| > 0$
3. $|xy| \leq p$

Proof sketch easy, using the pigeonhole principle

Example "corollary"

$L = \{ 0^n 1^n \mid n \in \mathbb{N}\}$ is nonregular.

Nonregular languages

every long enough word of a regular language can be pumped

Theorem (Pumping Lemma)

If L is a regular language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and

1. $xy^iz \in L$, for all $i \in \mathbb{N}$
2. $|y| > 0$
3. $|xy| \leq p$

Proof sketch easy, using the pigeonhole principle

Example "corollary"

$L = \{ 0^n1^n \mid n \in \mathbb{N}\}$ is nonregular.

Note the logical structure!

Context-free Grammars
and
Push-down Automata

Context-free Grammars

Informal example

$\Sigma = \{0,1\}$

G_1: $S \rightarrow 0S1, S \rightarrow A, A \rightarrow \varepsilon$

Context-free Grammars

alphabet (terminals)

Informal example

$\Sigma = \{0,1\}$

G_1: $S \to 0S1, S \to A, A \to \varepsilon$

Context-free Grammars

alphabet (terminals)

Informal example

S, A are variables (nonterminals)

$\Sigma = \{0,1\}$

G_1: $S \rightarrow 0S1, S \rightarrow A, A \rightarrow \varepsilon$

Context-free Grammars

alphabet (terminals)

Informal example

S, A are variables (nonterminals)

$\Sigma = \{0,1\}$

G_1: $S \rightarrow 0S1, S \rightarrow A, A \rightarrow \varepsilon$

S is initial nonterminal

Context-free Grammars

Informal example

$\Sigma = \{0,1\}$

G_1: $S \rightarrow 0S1, S \rightarrow A, A \rightarrow \varepsilon$

alphabet (terminals)

S, A are variables (nonterminals)

production rules "context-free!"

S is initial nonterminal

Context-free Grammars

alphabet (terminals)

Informal example

S, A are variables (nonterminals)

$\Sigma = \{0, 1\}$

G_1: $S \rightarrow 0S1$, $S \rightarrow A$, $A \rightarrow \varepsilon$

S is initial nonterminal

production rules "context-free!"

Generates the language $L(G_1) = \{0^n 1^n \mid n \in \mathbb{N}\}$

Context-free Grammars

alphabet (terminals)

Informal example

S, A are variables (nonterminals)

$\Sigma = \{0,1\}$

G_1: $S \to 0S1, S \to A, A \to \varepsilon$

S is initial nonterminal

production rules "context-free!"

Generates the language $L(G_1) = \{0^n 1^n \mid n \in \mathbb{N}\}$

context-free language

CFG

Definition

A context-free grammar G is a tuple G = (V, ∑, R, S) where

V is a finite set of variables (nonterminal symbols, nonterminals)
∑ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

CFG

Definition

A context-free grammar G is a tuple G = (V, Σ, R, S) where

V is a finite set of variables (nonterminal symbols, nonterminals)
Σ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

In the example G

CFG

Definition

A context-free grammar G is a tuple $G = (V, \Sigma, R, S)$ where

V is a finite set of variables (nonterminal symbols, nonterminals)
Σ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

In the example G

$G_1 = (V, \Sigma, R, S)$ for

CFG

Definition

A context-free grammar G is a tuple $G = (V, \Sigma, R, S)$ where

V is a finite set of variables (nonterminal symbols, nonterminals)
Σ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

In the example G

$G_1 = (V, \Sigma, R, S)$ for

$V = \{S, A\}$

CFG

Definition

A context-free grammar G is a tuple $G = (V, \Sigma, R, S)$ where

V is a finite set of variables (nonterminal symbols, nonterminals)
Σ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

In the example G

$G_1 = (V, \Sigma, R, S)$ for

$V = \{S, A\}$

$\Sigma = \{0, 1\}$

CFG

Definition

A context-free grammar G is a tuple G = (V, ∑, R, S) where

V is a finite set of variables (nonterminal symbols, nonterminals)
∑ is a finite alphabet (of terminal symbols, terminals)
R is a finite set of (production) rules, $R \subseteq V \times (\Sigma \cup V)^*$
S is the initial nonterminal, $S \in V$

In the example G

$G_1 = (V, \Sigma, R, S)$ for

$V = \{S, A\}$

$\Sigma = \{0, 1\}$

$R = \{ (S, 0S1), (S, A), (A, \varepsilon)\}$

CFG

Derivations

CFG

Derivations

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that

$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$

CFG

Derivations

uAv yields uwv

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$$

CFG

Derivations

u derives v

uAv yields uwv

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that

$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$

CFG

u derives v

Derivations

uAv yields uwv

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$$

Definition

The language generated by a context-free grammar
$G = (V, \Sigma, R, S)$ is

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

CFG

u derives v

Derivations

uAv yields uwv

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$$

Definition

The language generated by a context-free grammar $G = (V, \Sigma, R, S)$ is

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

all words consisting only of terminals, that can be derived from the initial nonterminal

CFG

u derives v

context-free language

Derivations

uAv yields uwv

Given $G = (V, \Sigma, R, S)$ we have $uAv \Rightarrow uwv$ for $u, v, w \in (\Sigma \cup V)^*$, $A \rightarrow w \in R$

and $u \Rightarrow^* v$ if $u = v$

or there exists a sequence $u_1, u_2, .., u_k$ for $k \geq 0$ such that
$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow .. \Rightarrow u_k \Rightarrow v$

Definition

The language generated by a context-free grammar $G = (V, \Sigma, R, S)$ is

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

all words consisting only of terminals, that can be derived from the initial nonterminal

Regular vs. CF languages

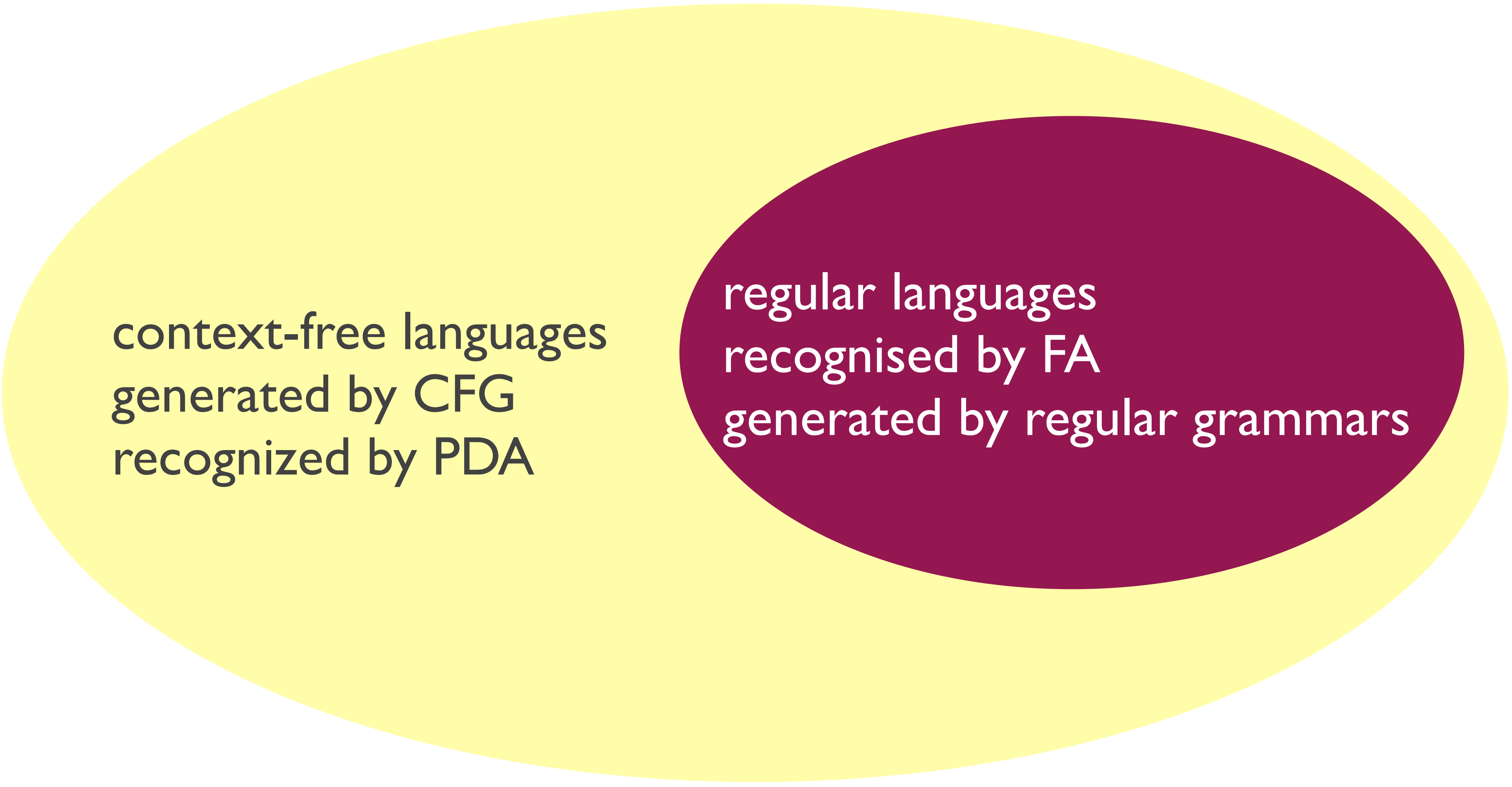
Theorem RL-CFL

The class of regular languages is contained in the class of context -free languages.

Regular vs. CF languages

Theorem RL-CFL

The class of regular languages is contained in the class of context -free languages.



Non-context-free languages

Non-context-free languages

Theorem (Pumping Lemma)

Non-context-free languages

every long enough word of a context-free language can be pumped at two places simultaneously

Theorem (Pumping Lemma)

Non-context-free languages

every long enough word of a context-free language can be pumped at two places simultaneously

Theorem (Pumping Lemma)

If L is a context-free language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $u, v, x, y, z \in \Sigma^*$ such that $w = uvxyz$ and

1. $uv^ixy^iz \in L$, for all $i \in \mathbb{N}$
2. $|vy| > 0$
3. $|vxy| \leq p$

Non-context-free languages

every long enough word of a context-free language can be pumped at two places simultaneously

Theorem (Pumping Lemma)

If L is a context-free language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $u, v, x, y, z \in \Sigma^*$ such that $w = uvxyz$ and

1. $uv^ixy^iz \in L$, for all $i \in \mathbb{N}$
2. $|vy| > 0$
3. $|vxy| \leq p$

Example "corollary"

$L = \{ a^n b^n c^n \mid n \in \mathbb{N}\}$ is non-context-free.

Non-context-free languages

every long enough word of a context-free language can be pumped at two places simultaneously

Theorem (Pumping Lemma)

If L is a context-free language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $u, v, x, y, z \in \Sigma^*$ such that $w = uvxyz$ and

1. $uv^ixy^iz \in L$, for all $i \in \mathbb{N}$
2. $|vy| > 0$
3. $|vxy| \leq p$

Proof sketch easy, using the pigeonhole principle

Example "corollary"

$L = \{ a^nb^nc^n \mid n \in \mathbb{N}\}$ is non-context-free.

Non-context-free languages

every long enough word of a context-free language can be pumped at two places simultaneously

Theorem (Pumping Lemma)

If L is a context-free language, then there is a number $p \in \mathbb{N}$ (the pumping length) such that for any $w \in L$ with $|w| \geq p$, there exist $u, v, x, y, z \in \Sigma^*$ such that $w = uvxyz$ and

1. $uv^ixy^iz \in L$, for all $i \in \mathbb{N}$
2. $|vy| > 0$
3. $|vxy| \leq p$

Proof sketch easy, using the pigeonhole principle

Example "corollary"

$L = \{ a^nb^nc^n \mid n \in \mathbb{N}\}$ is non-context-free.

Note the logical structure!

Properties of CF languages

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

> but not under intersection!

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

but not under intersection!

and not under complement!

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

but not under intersection!

Theorem CF2

The class of regular languages is closed under concatenation

and not under complement!

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

but not under intersection!

Theorem CF2

The class of regular languages is closed under concatenation

and not under complement!

Theorem CF3

The class of regular languages is closed under Kleene star

Properties of CF languages

Theorem CF1

The class of regular languages is closed under union

but not under intersection!

Theorem CF2

The class of regular languages is closed under concatenation

and not under complement!

Theorem CF3

The class of regular languages is closed under Kleene star

Theorem CF4

The intersection of a regular language and a context-free language is context-free