COL 352 Introduction to Automata and Theory of Computation

Nikhil Balaji

Bharti 420
Indian Institute of Technology, Delhi
nbalaji@cse.iitd.ac.in

March 1, 2023

Lecture 16: Context-Free Languages

Recap

Definition

A non-deterministic pushdown automaton (NPDA)
$A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$, where

Q:	set of states	Σ:	input alphabet
Γ:	stack alphabet	q_0:	start state
$\perp$:	start symbol	F:	set of final states

$\delta \subseteq Q \times \Sigma \times \Gamma \times Q \times \Gamma^*$.

Understanding δ

For $q \in Q, a \in \Sigma$ and $X \in \Gamma$, if $\delta(q, a, X) = (p, \gamma)$,
then p is the new state and γ replaces X in the stack.

if $\gamma = \epsilon$ then X is popped.
if $\gamma = X$ then X stays unchanges on the top of the stack.
if $\gamma = \gamma_1\gamma_2 \ldots \gamma_k$ then X is replaced by γ_k
and $\gamma_1\gamma_2 \ldots \gamma_{k-1}$ are pushed on top of that.

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then $\forall w \in \Sigma^*$ and $\gamma' \in \Gamma^*$

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then $\forall w \in \Sigma^*$ and $\gamma' \in \Gamma^*$,

$$(q, a \cdot w, X\gamma') \vdash (p, w, \gamma \cdot \gamma')$$

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then $\forall w \in \Sigma^*$ and $\gamma' \in \Gamma^*$,

$$(q, a \cdot w, X\gamma') \ \vdash \ (p, w, \gamma \cdot \gamma')$$

Let I, J are two configurations of A.

We say that $I \vdash^k J$ iff $\exists I'$ such that $I \vdash I'$

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \bot, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then $\forall w \in \Sigma^*$ and $\gamma' \in \Gamma^*$,

$$(q, a \cdot w, X\gamma') \;\vdash\; (p, w, \gamma \cdot \gamma')$$

Let I, J are two configurations of A.

We say that $I \vdash^k J$ iff $\exists I'$ such that $I \vdash I'$ and $I' \vdash^{k-1} J$.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \bot) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if
$(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w$ is accepted by $A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \epsilon)$

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \epsilon)$, where $q \in Q$.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \epsilon)$, where $q \in Q$. acceptance by an empty stack.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \epsilon)$, where $q \in Q$. acceptance by an empty stack.

Equivalence of two notions

Equivalence of two notions

Acceptance by PDA

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA.

- If L is the language accepted of P by final state, there exists PDA P' such that its language accepted by empty stack is L.
- If L is the language accepted of P by empty stack , there exists PDA P' such that its language accepted by final state is L.

Equivalence of two notions

Acceptance by PDA

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA.

- If L is the language accepted of P by final state, there exists PDA P' such that its language accepted by empty stack is L.
- If L is the language accepted of P by empty stack , there exists PDA P' such that its language accepted by final state is L.

Proof: Exercise (Kozen Supplementary lecture E)

Deterministic PDA

Definition

A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a DPDA if for each $q \in Q$ and $X \in \Gamma$

- $|\delta(q, a, X)| \leq 1$ for each $a \in \Sigma \cup \{\epsilon\}$
- if $|\delta(q, a, X)| = 1$ for some $a \in \Sigma$, then $|\delta(q, \epsilon, X)| = 0$

Context-free languages

- Are context-free languages closed under union? intersection? complement? concatenation?

Context-free languages

- Are context-free languages closed under union? intersection? complement? concatenation?
- Recall: for regular languages we had regular expressions to write rules.
- Are there similar rules for languages beyond regular?

Context-free languages

- Are context-free languages closed under union? intersection? complement? concatenation?
- Recall: for regular languages we had regular expressions to write rules.
- Are there similar rules for languages beyond regular?
- Are there non-context free languages?

Context-free languages

- Are context-free languages closed under union? intersection? complement? concatenation?
- Recall: for regular languages we had regular expressions to write rules.
- Are there similar rules for languages beyond regular?
- Are there non-context free languages?

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

$S \to 0S0$.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

$S \to 0S0$.

$S \to 1S1$.

Context-free grammars

Inductive definition of PAL.

ϵ is in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

$S \to 0S0$.

$S \to 1S1$.

Context-free grammars

PAL' $= \left\{ w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\} \right\}$

Context-free grammars

PAL' $= \left\{ w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\} \right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$

Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \to \epsilon$.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \rightarrow \epsilon$.

$S \rightarrow 0$.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$

Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \rightarrow \epsilon$.

$S \rightarrow 0$.

$S \rightarrow 1$.

$S \rightarrow 0S0$.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$

Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \rightarrow \epsilon$.

$S \rightarrow 0$.

$S \rightarrow 1$.

$S \rightarrow 0S0$.

$S \rightarrow 1S1$.

Context-free grammars

PAL' = $\left\{w \cdot c \cdot w^R \mid w \in \{0,1\}^*, c \in \{0,1,\epsilon\}\right\}$
Inductive definition of PAL'.

$\epsilon, 0, 1$ is in PAL'.

If w is in PAL' then $0 \cdot w \cdot 0 \in$ PAL'.

If w is in PAL' then $1 \cdot w \cdot 1 \in$ PAL'.

Context-free grammar for PAL'.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

Formal Grammars informally..

A (formal) grammar consists of

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \rightarrow \psi$$

   where ϕ and ψ are strings of symbols.

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \to \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \to \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \to \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.
4. Other symbols stand for "phrases" and are called non-terminal vocabulary.

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \to \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.
4. Other symbols stand for "phrases" and are called non-terminal vocabulary.

Given such a grammar, a valid sentence can be generated by

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form
$$\phi \to \psi$$
where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.
4. Other symbols stand for "phrases" and are called non-terminal vocabulary.

Given such a grammar, a valid sentence can be generated by

1. Starting from the initial symbol S,

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \rightarrow \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.
4. Other symbols stand for "phrases" and are called non-terminal vocabulary.

Given such a grammar, a valid sentence can be generated by

1. Starting from the initial symbol S,
2. Applying one of the rewriting rules to form a new string ϕ by applying a rule $S \rightarrow \phi_1$,

Formal Grammars informally..

A (formal) grammar consists of

1. A finite set of rewriting rules of the form

$$\phi \to \psi$$

   where ϕ and ψ are strings of symbols.
2. A special "initial" symbol S (S standing for sentence)
3. A finite set of symbols stand for "words" of the language called terminal vocabulary.
4. Other symbols stand for "phrases" and are called non-terminal vocabulary.

Given such a grammar, a valid sentence can be generated by

1. Starting from the initial symbol S,
2. Applying one of the rewriting rules to form a new string ϕ by applying a rule $S \to \phi_1$,
3. and apply another rule to form a new string ϕ_2 and so on,
4. until we reach a string ϕ_n that consists only of terminal symbols.

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary)

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon$.

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,

$S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,

$S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,

$S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,

$S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

$G_{pal} = (V, T, P, S_0)$ such that

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

V is a set of variables (non-terminals/vocabulary),

T is a set of terminal symbols or the alphabet,

P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,

$S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

$G_{pal} = (V, T, P, S_0)$ such that

$V = \{S\}$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

$G_{pal} = (V, T, P, S_0)$ such that

$V = \{S\}$,

$T = \{0, 1\}$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

$G_{pal} = (V, T, P, S_0)$ such that

$V = \{S\}$,

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

$P = \{S \to \epsilon, S \to 0, S \to 1, S \to 0S0, S \to 1S1\}$

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables (non-terminals/vocabulary),
- T is a set of terminal symbols or the alphabet,
- P is a set of (rewriting rules) productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start ("sentence") symbol.

Example: Grammar for PAL.

$S \to \epsilon.$

$S \to 0.$

$S \to 1.$

$S \to 0S0.$

$S \to 1S1.$

$G_{pal} = (V, T, P, S_0)$ such that

$V = \{S\},$

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

$P = \{S \to \epsilon, S \to 0, S \to 1, S \to 0S0, S \to 1S1\},$

$S_0 = S.$

Context-free grammars

Example:

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \rightarrow \epsilon \mid aSb$$

Example:

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example:

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

$$S'' \to a \mid aS''b \mid aS''$$

Example:

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

$$S'' \to a \mid aS''b \mid aS''$$

Example: $L = \{a^i b^j \mid i \neq j\}$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

$$S'' \to a \mid aS''b \mid aS''$$

Example: $L = \{a^i b^j \mid i \neq j\}$

$$S \to S' \mid S''$$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

$$S'' \to a \mid aS''b \mid aS''$$

Example: $L = \{a^i b^j \mid i \neq j\}$

$$S \to S' \mid S''$$

$$S' \to b \mid aS'b \mid S'b$$

Context-free grammars

Example: $L_{a,b} = \{a^n b^n \mid n \geq 0\}$

$$S \to \epsilon \mid aSb$$

Example: $L' = \{a^i b^j \mid i < j\}$

$$S' \to b \mid aS'b \mid S'b$$

Example: $L'' = \{a^i b^j \mid i > j\}$

$$S'' \to a \mid aS''b \mid aS''$$

Example: $L = \{a^i b^j \mid i \neq j\}$

$$S \to S' \mid S''$$

$$S' \to b \mid aS'b \mid S'b$$

$$S'' \to a \mid aS''b \mid aS''.$$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^0 \alpha$.

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^0 \alpha$.

For all $\alpha, \beta, \gamma \in (V \cup T)^*$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^0 \alpha$.

For all $\alpha, \beta, \gamma \in (V \cup T)^*$,

if $\alpha \Rightarrow^{k-1} \beta$ and $\beta \Rightarrow \gamma$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^0 \alpha$.

For all $\alpha, \beta, \gamma \in (V \cup T)^*$,

if $\alpha \Rightarrow^{k-1} \beta$ and $\beta \Rightarrow \gamma$ then $\alpha \Rightarrow^k \gamma$.

For all $\alpha, \beta \in (V \cup T)^*$

Derivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^0 \alpha$.

For all $\alpha, \beta, \gamma \in (V \cup T)^*$,

if $\alpha \Rightarrow^{k-1} \beta$ and $\beta \Rightarrow \gamma$ then $\alpha \Rightarrow^k \gamma$.

For all $\alpha, \beta \in (V \cup T)^*$, we say that $\alpha \Rightarrow^* \beta$, if $\exists k \geq 0$ s.t. $\alpha \Rightarrow^k \beta$.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \; w \in L(G_{Pal})$ if and only if $w = w^R$.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \; w \in L(G_{Pal})$ if and only if $w = w^R$.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \; w \in L(G_{Pal})$ if and only if $w = w^R$.

Proof.

$(\Leftarrow)$: If $w = w^R$, $w \in L(G_{Pal})$

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \ w \in L(G_{Pal})$ if and only if $w = w^R$.

Proof.

$(\Leftarrow)$: If $w = w^R$, $w \in L(G_{Pal})$
By Induction on $|w|$.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

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

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \ w \in L(G_{Pal})$ if and only if $w = w^R$.

Proof.

($\Leftarrow$): If $w = w^R$, $w \in L(G_{Pal})$
By Induction on $|w|$.
($\Rightarrow$): If $w \in L(G_{Pal})$ then $w = w^R$.
By Induction on $|w|$. □

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid \text{ either } i \neq j \text{ or } j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$
$S' \to b \mid aS'b \mid S'b$

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$
$S' \to b \mid aS'b \mid S'b$
$S'' \to a \mid aS''b \mid aS''$.
$C \to cC \mid \epsilon$

Now we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid j \neq k\}$.

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$
$S' \to b \mid aS'b \mid S'b$
$S'' \to a \mid aS''b \mid aS''$.
$C \to cC \mid \epsilon$

Now we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid j \neq k\}$.

$S \to AS' \mid AS''$

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$
$S' \to b \mid aS'b \mid S'b$
$S'' \to a \mid aS''b \mid aS''$.
$C \to cC \mid \epsilon$

Now we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid j \neq k\}$.

$S \to AS' \mid AS''$
$S' \to c \mid bS'c \mid S'c$

Examples

Give context-free grammars for the following languages.

$\{a^i \cdot b^j \cdot c^k \mid$ either $i \neq j$ or $j \neq k\}$.

First we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid i \neq j\}$.

$S \to S'C \mid S''C$
$S' \to b \mid aS'b \mid S'b$
$S'' \to a \mid aS''b \mid aS''$.
$C \to cC \mid \epsilon$

Now we give a grammar for $\{a^i \cdot b^j \cdot c^k \mid j \neq k\}$.

$S \to AS' \mid AS''$
$S' \to c \mid bS'c \mid S'c$
$S'' \to b \mid bS''c \mid bS''$.
$C \to aA \mid \epsilon$

Now simply take the union.

Exercises

- Give a grammar for the language of all valid regular expressions.
- Give a grammar for any regular language (given as a DFA).

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \rightarrow BC$$

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \to BC$$

$$A \to a$$

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \to BC$$

$$A \to a$$

where $a \in T$, $A, B, C \in V$, neither B nor C is the start variable, i.e. start variabe does not appear on the right of any rule.

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \to BC$$

$$A \to a$$

where $a \in T$, $A, B, C \in V$, neither B nor C is the start variable, i.e. start variabe does not appear on the right of any rule. Moreover, epsilon does not appear on the right of any rule except as $S \to \epsilon$.

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \to BC$$

$$A \to a$$

where $a \in T$, $A, B, C \in V$, neither B nor C is the start variable, i.e. start variabe does not appear on the right of any rule. Moreover, epsilon does not appear on the right of any rule except as $S \to \epsilon$.

Lemma

Any context-free grammar G can be converted into another context-free grammar G'

Chomsky normal form

Definition

A context-free grammar is said to be in Chomsky normal form if every rule is of the form

$$A \to BC$$

$$A \to a$$

where $a \in T$, $A, B, C \in V$, neither B nor C is the start variable, i.e. start variabe does not appear on the right of any rule. Moreover, epsilon does not appear on the right of any rule except as $S \to \epsilon$.

Lemma

Any context-free grammar G can be converted into another context-free grammar G' such that $L(G) = L(G')$ and G' is in the Chomsky normal form.