CS310 Automata Theory – 2016-2017

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Lecture 27: Turing machines, computability
March 27, 2017

At the end of last class

Introduction to Turing machines

Undecidability of the following languages:

$A_{TM} = \{(M, w) \mid M \text{ accepts } w\}$.

At the end of last class

Introduction to Turing machines

Undecidability of the following languages:

$A_{TM} = \{(M, w) \mid M \text{ accepts } w\}$.

$\text{Halt} = \{(M, w) \mid M \text{ hants on } w\}$.

At the end of last class

Introduction to Turing machines

- Undecidability of the following languages:
  - $A_{TM} = \{(M, w) \mid M \text{ accepts } w\}$.
  - $\text{Halt} = \{(M, w) \mid M \text{ hants on } w\}$.
  - $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$.

At the end of last class

Introduction to Turing machines

- Undecidability of the following languages:
  - $A_{TM} = \{(M, w) \mid M \text{ accepts } w\}$.
  - $\text{Halt} = \{(M, w) \mid M \text{ hants on } w\}$.
  - $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$.
  - $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$.

At the end of last class

Introduction to Turing machines

Undecidability of the following languages:

$A_{TM} = \{(M, w) \mid M \text{ accepts } w\}$.

Halt $= \{(M, w) \mid M$ hants on $w\}$.

$E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$.

$EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$.

Started with trying to prove $REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$.

Undecidability of Halt

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable.

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w).

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject,

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject, else do as per M on w.

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject, else do as per M on w.

$\mathcal{A}$ accepts (M, w) if M accepts w

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject, else do as per M on w.

$\mathcal{A}$ accepts (M, w) if M accepts w and rejects it if either M rejects w

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject, else do as per M on w.

$\mathcal{A}$ accepts (M, w) if M accepts w and rejects it if either M rejects w or M loops forever on w.

Undecidability of Halt

Reducing A_{TM} to another problem to prove undecidibility.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let $\mathcal{H}$ be the TM deciding Halt.

$\mathcal{A}$: Run $\mathcal{H}$ on (M, w). If it rejects then reject, else do as per M on w.

$\mathcal{A}$ accepts (M, w) if M accepts w and rejects it if either M rejects w or M loops forever on w.

$\mathcal{H}$ decides Halt if and only if $\mathcal{A}$ decides A_{TM}.

The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$*, is undecidable.*

The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$, is undecidable.

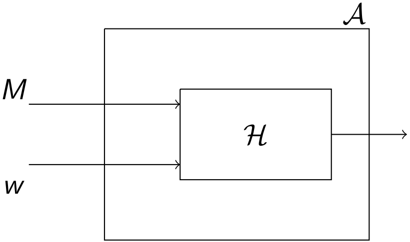
Another way to describe the same proof.

The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$, is undecidable.

Another way to describe the same proof.

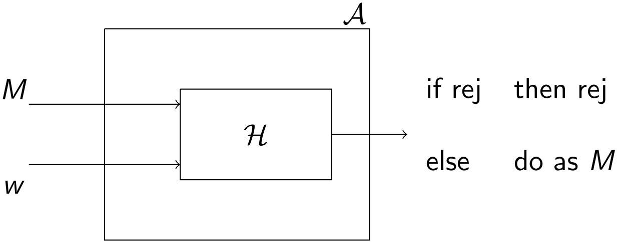


The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$, is undecidable.

Another way to describe the same proof.

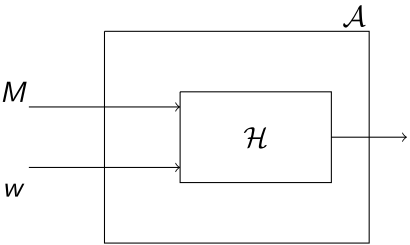


The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$, is undecidable.

Another way to describe the same proof.



if rej then rej

else do as M

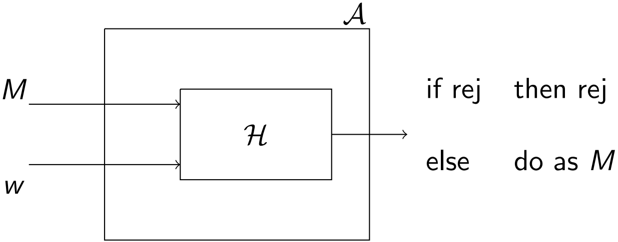
If Halt is decidable then $\mathcal{A}$ decides A_{TM}

The halting problem

Lemma

The halting problem, $Halt = \{(M, w) \mid M \text{ halts on } w\}$, is undecidable.

Another way to describe the same proof.



If Halt is decidable then $\mathcal{A}$ decides A_{TM}, which is a contradiction.

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$, is undecidable.

Assume for the sake of contradiction that it is decidable.

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Let $T'_{M,w}$ be as follows:

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Let $T'_{M,w}$ be as follows:

On input x

{

if $w \neq x$ then reject

else do as per M

}

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Let $T'_{M,w}$ be as follows:

On input x

{

if $w \neq x$ then reject

else do as per M

}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \emptyset & \text{otherwise} \end{cases}$$

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Let $T'_{M,w}$ be as follows:

On input x

{

if $w \neq x$ then reject

else do as per M

}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \varnothing & \text{otherwise} \end{cases}$$

Let A be as follows:

Emptiness problem for TM

Lemma

The emptiness problem for TMs, $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$, is undecidable.

Assume for the sake of contradiction that it is decidable. Let T be a machine that decides E_{TM}.

Let $T'_{M,w}$ be as follows:

On input x

{

if $w \neq x$ then reject

else do as per M

}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \emptyset & \text{otherwise} \end{cases}$$

Let A be as follows:

On input M, w

{

Create machine $T'_{M,w}$.

If T on $\langle T'_{M,w} \rangle$ rejects

then accept

else reject

}

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, *is undecidable.*

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Let M_1 be a machine that rejects all strings. That is, $L(M_1) = \emptyset$.

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Let M_1 be a machine that rejects all strings. That is, $L(M_1) = \emptyset$.

Given a machine M_2 as an input, use M to check whether $L(M_2) = L(M_1)$

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Let M_1 be a machine that rejects all strings. That is, $L(M_1) = \emptyset$.

Given a machine M_2 as an input, use M to check whether $L(M_2) = L(M_1)$, i.e. to check whether $L(M_2) = \emptyset$ or not.

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Let M_1 be a machine that rejects all strings. That is, $L(M_1) = \varnothing$.

Given a machine M_2 as an input, use M to check whether $L(M_2) = L(M_1)$, i.e. to check whether $L(M_2) = \varnothing$ or not.

This implies that if EQ_{TM} is decidable then E_{TM} is decidable.

Equality for TM

Lemma

The equality problem for TMs, $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$, is undecidable.

Assume for the sake of contradiction that EQ_{TM} is decidable. Let M be the TM for it.

Let M_1 be a machine that rejects all strings. That is, $L(M_1) = \varnothing$.

Given a machine M_2 as an input, use M to check whether $L(M_2) = L(M_1)$, i.e. to check whether $L(M_2) = \varnothing$ or not.

This implies that if EQ_{TM} is decidable then E_{TM} is decidable.

But from the previous result we know that E_{TM} is undecidable.

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Let $R'_{M,w}$ be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Let $R'_{M,w}$ be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an $R'_{M,w}$ we can design A as a follows.

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Let $R'_{M,w}$ be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an $R'_{M,w}$ we can design A as a follows.

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Let $R'_{M,w}$ be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an $R'_{M,w}$ we can design A as a follows.

Let A be as follows:

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R is a TM that decides REG_{TM}.

Let $R'_{M,w}$ be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an $R'_{M,w}$ we can design A as a follows.

Let A be as follows:

On input M, w

{

Create machine $R'_{M,w}$.

If R on $\langle R'_{M,w}\rangle$ accepts

then accept

else reject

}

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R be a TM that decides REG_{TM}.

Let $R'_{M,w}$ be as follows:

Regularity checking

Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R be a TM that decides REG_{TM}.

Let $R'_{M,w}$ be as follows:

On input x

$\{$

if $x = 0^n 1^n$

then accept

else run M on w and

if M acc w then acc

else rej

$\}$

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R be a TM that decides REG_{TM}.

Let $R'_{M,w}$ be as follows:

On input x

{

if $x = 0^n1^n$

then accept

else run M on w and

if M acc w then acc

else rej

}

Let A be as follows:

Regularity checking

Lemma

$REG_{TM} = \{\langle M\rangle \mid L(M)$ is regular$\}$ is undecidable.

Assume for the sake of contradiction that a TM R be a TM that decides REG_{TM}.

Let $R'_{M,w}$ be as follows:

On input x

{

if $x = 0^n1^n$

then accept

else run M on w and

  if M acc w then acc
  else rej

}

Let A be as follows:

On input M, w

{

Create machine $R'_{M,w}$.

If R on $\langle R'_{M,w}\rangle$ accepts

then accept

else reject

}

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M\rangle \mid M$ is a PDA and $L(M) = \Sigma^*\}$ is undecidable.

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M \rangle \mid M$ is a PDA and $L(M) = \Sigma^*\}$ is undecidable.

Proof Strategy

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M\rangle \mid M$ is a PDA and $L(M) = \Sigma^*\}$ is undecidable.

Proof Strategy

$$\text{Input } (M, w) \quad \longrightarrow \quad N$$

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M \rangle \mid M$ is a PDA and $L(M) = \Sigma^*\}$ is undecidable.

Proof Strategy

$$\text{Input } (M, w) \quad \longrightarrow \quad N$$

$$\text{if } w \in L(M) \quad \longrightarrow \quad \exists x \in \Sigma^*, \text{ s.t. } x \notin L(N)$$

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M \rangle \mid M$ is a PDA and $L(M) = \Sigma^*\}$ is undecidable.

Proof Strategy

$$\text{Input } (M, w) \longrightarrow N$$

$$\text{if } w \in L(M) \longrightarrow \exists x \in \Sigma^*, \text{ s.t. } x \notin L(N)$$

$$\text{if } w \notin L(M) \longrightarrow L(N) = \Sigma^*$$

Universality of CFLs

Lemma

$ALL_{CFL} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) = \Sigma^*\}$ *is undecidable.*

Proof Strategy

$$\text{Input } (M, w) \longrightarrow N$$

$$\text{if } w \in L(M) \longrightarrow \exists x \in \Sigma^*, \text{ s.t. } x \notin L(N)$$

$$\text{if } w \notin L(M) \longrightarrow L(N) = \Sigma^*$$

We will do the proof on Thursday (after the Tutorial 10 is solved).

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

$\{M \mid L(M) \text{ is context-free}\}$.

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

$\{M \mid L(M) \text{ is context-free}\}$.

$\{M \mid L(M) = \varnothing\}$.

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

$\{M \mid L(M) \text{ is context-free}\}$.

$\{M \mid L(M) = \varnothing\}$.

The following languages are decidable.

$\{M \mid M \text{ has more than 10 states}\}$.

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

$\{M \mid L(M) \text{ is context-free}\}$.

$\{M \mid L(M) = \varnothing\}$.

The following languages are decidable.

$\{M \mid M \text{ has more than 10 states}\}$.

$\{M \mid M \text{ never moves left on any input}\}$.

Rice's theorem

The following languages are undecidable.

$\{M \mid L(M) \text{ is regular}\}$.

$\{M \mid L(M) \text{ is context-free}\}$.

$\{M \mid L(M) = \varnothing\}$.

The following languages are decidable.

$\{M \mid M \text{ has more than 10 states}\}$.

$\{M \mid M \text{ never moves left on any input}\}$.

Rice's theorem: A systematic way of proving undecidability of languages.

Property P

Definition

A property P is simply a subset of Turing recognizable languages.

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages.

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. ✓

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. $\checkmark$

Set of context-free languages.

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. ✓

Set of context-free languages. ✓

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. ✓

Set of context-free languages. ✓

$\{\emptyset\}$.

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. ✓

Set of context-free languages. ✓

$\{\emptyset\}$. ✓

Property P

Definition

A property P is simply a subset of Turing recognizable languages. We say that a language L satisfies a property P, if $L \in P$.

Examples

Set of regular languages. ✓

Set of context-free languages. ✓

$\{\varnothing\}$. ✓