Automata & Complexity

Jörg Endrullis

Vrije Universiteit Amsterdam

2018

Decidability

A **decision problem** P is a language $P \subseteq \Sigma^*$.

The problem P is called

- **decidable** if the P is recursive, otherwise **undeciable**,
- **semidecidable** if the P is recursively enumerable.

Decidability

A **decision problem** P is a language $P \subseteq \Sigma^*$.

The problem P is called

- **decidable** if the P is recursive, otherwise **undeciable**,
- **semidecidable** if the P is recursively enumerable.

Decidable problem:

- algorithm that always halts
- always answers yes or no

Decidability

A **decision problem** P is a language $P \subseteq \Sigma^*$.

The problem P is called

- **decidable** if the P is recursive, otherwise **undeciable**,
- **semidecidable** if the P is recursively enumerable.

Decidable problem:

- algorithm that always halts
- always answers yes or no

Semidecidable problem:

- algorithm halts (eventually) it the answer is yes ($w \in P$),
- may or may not halt if the answer is no ($w \notin P$).

(Problem: one cannot know how long to wait for an answer.)

Decidability

A decision problem P is decidable if

- P is semidecidable, and
- $\overline{P}$ is semidecidable.

Decidability

A decision problem P is decidable if

- P is semidecidable, and
- $\overline{P}$ is semidecidable.

The following question is undecidable, but semidecidable:

Halting problem

Does TM M reach a halting state for input w? (Input: M and w.)

Decidability

A decision problem P is decidable if

- P is semidecidable, and
- $\overline{P}$ is semidecidable.

The following question is undecidable, but semidecidable:

Halting problem

Does TM M reach a halting state for input w? (Input: M and w.)

(Semidecidable: execute M on w and wait.)

Decidability

A decision problem P is decidable if

- P is semidecidable, and
- $\overline{P}$ is semidecidable.

The following question is undecidable, but semidecidable:

Halting problem

Does TM M reach a halting state for input w? (Input: M and w.)

(Semidecidable: execute M on w and wait.)

The following question not decidable and not semidecidable:

Universal halting problem

Does TM M reach a halting state on all $w \in \Sigma^*$? (Input: M.)

(The complement is also not semidecidable.)

The Halting Problem (1936)

The **halting problem** is: given

- a deterministic Turing machine M and
- a word x,

does M reach a halting state when started with input x?

The Halting Problem (1936)

The **halting problem** is: given

- a deterministic Turing machine M and
- a word x,

does M reach a halting state when started with input x?

The halting problem can be viewed as a language H

$$H = \{ (M, x) \mid M \text{ reaches a halting state on input } x \}$$

M is an encoding of a deterministic Turing machine as a word.

The Halting Problem (1936)

The **halting problem** is: given

- a deterministic Turing machine M and
- a word x,

does M reach a halting state when started with input x?

The halting problem can be viewed as a language H

$$H = \{ (M, x) \mid M \text{ reaches a halting state on input } x \}$$

M is an encoding of a deterministic Turing machine as a word.

Theorem

The halting problem H is undecidable.

(The language H is not recursive.)

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

Then every recursively enumerable language was recursive:

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

Then every recursively enumerable language was recursive:

Let M be a deterministic Turing machine and x a word.

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

Then every recursively enumerable language was recursive:

Let M be a deterministic Turing machine and x a word.

We can decide $x \in L(M)$ as follows:

- If according to $\mathcal{H}$, M does not halt on x, then $x \notin L(M)$.
- If according to $\mathcal{H}$, M halts on x, then execute M on x to see whether $x \in L(M)$.

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

Then every recursively enumerable language was recursive:

Let M be a deterministic Turing machine and x a word.

We can decide $x \in L(M)$ as follows:

- If according to $\mathcal{H}$, M does not halt on x, then $x \notin L(M)$.
- If according to $\mathcal{H}$, M halts on x, then execute M on x to see whether $x \in L(M)$.

The algorithm always terminates, so $L(M)$ is recursive.

The Halting Problem is Undecidable (Proof 1)

Proof.

Assume that there was a deterministic TM $\mathcal{H}$ that, given (M, x) decides whether M halts on x (that is, $(M, x) \in H$).

Then every recursively enumerable language was recursive:

Let M be a deterministic Turing machine and x a word.

We can decide $x \in L(M)$ as follows:

- If according to $\mathcal{H}$, M does not halt on x, then $x \notin L(M)$.
- If according to $\mathcal{H}$, M halts on x, then execute M on x to see whether $x \in L(M)$.

The algorithm always terminates, so $L(M)$ is recursive.

Contradiction: not every recursively enumerable language is recursive. □

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program *T* with the behaviour:

- input: a program *M*
- output: *yes* if *M* terminates on input *M*, *no* otherwise

program *T*

```
p = read input;
      ⋮
print result;
```

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program *T* with the behaviour:

- input: a program *M*
- output: *yes* if *M* terminates on input *M*, *no* otherwise

program *T*:

```
p = read input;
      ⋮
print result;
```

program *T′* (program *T* followed by):

```
if result = yes
then  loop forever
else  terminate
```

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then  loop forever
else  terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then  loop forever
else  terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then  loop forever
else  terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then  loop forever
else  terminate
```

(both parts together: program T')

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever Contradiction

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then   loop forever
else   terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever Contradiction
- if the result is no, then T' terminates

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

program T' = program T followed by:

```
if result = yes
then   loop forever
else   terminate
```

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever   Contradiction
- if the result is no, then T' terminates   Contradiction

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then  loop forever
else  terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever   Contradiction
- if the result is no, then T' terminates   Contradiction

Thus T cannot exist!

The Halting Problem is Undecidable (Proof 2)

Assume there would be a program T with the behaviour:

- input: a program M
- output: *yes* if M terminates on input M, *no* otherwise

program T:

```
p = read input;
      ⋮
print result;
```

```
if result = yes
then   loop forever
else   terminate
```

program T' (both parts together)

What happens if we run T' with input T'?

- initial part T decides whether T' terminates on input T'
- if the result is yes, then T' runs forever   Contradiction
- if the result is no, then T' terminates   Contradiction

Thus T cannot exist! The halting problem is undecidable!

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable. We decide $x \in L(M)$!

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable. We decide $x \in L(M)$!

For a word x, we construct a Turing machine M_x with

$$L(M_x) = \emptyset \quad \text{if } x \notin L \qquad\qquad L(M_x) = L_0 \quad \text{if } x \in L$$

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable. We decide $x \in L(M)$!

For a word x, we construct a Turing machine M_x with

$$L(M_x) = \emptyset \quad \text{if } x \notin L \qquad\qquad L(M_x) = L_0 \quad \text{if } x \in L$$

M_x accepts y if $x \in L$ and $y \in L_0$.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable. We decide $x \in L(M)$!

For a word x, we construct a Turing machine M_x with

$$L(M_x) = \emptyset \quad \text{if } x \notin L \qquad\qquad L(M_x) = L_0 \quad \text{if } x \in L$$

M_x accepts y if $x \in L$ and $y \in L_0$. Then $x \notin L \iff P(L(M_x))$.

Theorem of Rice (1951)

A property of a class K is **trivial** if it holds for **all** or **no** $k \in K$.

Theorem of Rice

Every **non-trivial** property P of recursively enumerable languages is undecidable.

Proof.

Assume that $P(\emptyset)$ (if not, take $\neg P$).

Let L_0 be a recursively enumerable language with $\neg P(L_0)$.

Let L be recursively enumerable. We decide $x \in L(M)$!

For a word x, we construct a Turing machine M_x with

$$L(M_x) = \emptyset \quad \text{if } x \notin L \qquad\qquad L(M_x) = L_0 \quad \text{if } x \in L$$

M_x accepts y if $x \in L$ and $y \in L_0$. Then $x \notin L \iff P(L(M_x))$.

Contradiction: decidability of $P \implies L$ recursive. □

Theorem of Rice: Example

For recursively enumerable languages L, the following questions are undecidable:

1. Is $a \in L$?
2. Is L finite?

Post Correspondence Problem (1946)

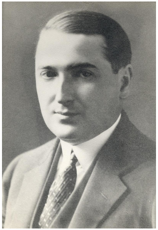
Post Correspondence Problem (PCP)

Given n pairs of words:

$$(w_1, v_1), \ \ldots, \ (w_n, v_n)$$

Are there indices $i_1, i_2 \ldots, i_k$ ($k \geq 1$) s.t.

$$w_{i_1} w_{i_2} \cdots w_{i_k} \quad = \quad v_{i_1} v_{i_2} \cdots v_{i_k} \ ?$$



Emil Post
(1897-1954)

Post Correspondence Problem (1946)

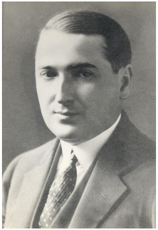
Post Correspondence Problem (PCP)

Given n pairs of words:

$$(w_1, v_1), \ \ldots, \ (w_n, v_n)$$

Are there indices $i_1, i_2 \ldots, i_k$ ($k \geq 1$) s.t.

$$w_{i_1} w_{i_2} \cdots w_{i_k} \quad = \quad v_{i_1} v_{i_2} \cdots v_{i_k} \ \ ?$$



Emil Post
(1897-1954)

Exercise

Find a solution for

$$(w_1, v_1) = (01, 100)$$
$$(w_2, v_2) = (1, 011)$$
$$(w_3, v_3) = (110, 1)$$

Modified Post Correspondence Problem

We will show that the PCP is undecidable.

Modified Post Correspondence Problem

We will show that the PCP is undecidable.

We first prove that the **modified PCP (MPCP)** is undecidable.

Modified Post Correspondence Problem

We will show that the PCP is undecidable.

We first prove that the **modified PCP (MPCP)** is undecidable.

Modified PCP (MPCP)

Given n pairs of words:

$$(w_1, v_1),\ \ldots,\ (w_n, v_n)$$

Are there indices $i_1, i_2 \ldots, i_k$ ($k \geq 1$) such that

$$w_1\, w_{i_1}\, w_{i_2} \cdots w_{i_k} \quad = \quad v_1\, v_{i_1}\, v_{i_2} \cdots v_{i_k} \ ?$$

Modified Post Correspondence Problem

Theorem

The modified PCP is undecidable.

Proof.

Let $G = (V, T, S, P)$ be an unrestricted grammar.

Modified Post Correspondence Problem

Theorem

The modified PCP is undecidable.

Proof.

Let $G = (V, T, S, P)$ be an unrestricted grammar.

We define (where F and E are fresh):

$$
\begin{array}{rlrll}
w_1 = & F & v_1 = & FS \Rightarrow & \\
w_2 = & \Rightarrow wE & v_2 = & E & \\
\vdots & x & \vdots & y & (x \to y \in P) \\
 & a & & a & (a \in T) \\
 & A & & A & (A \in V) \\
 & \Rightarrow & & \Rightarrow &
\end{array}
$$

This MPCP has a solution $\iff w \in L(G)$.

Modified Post Correspondence Problem

Theorem

The modified PCP is undecidable.

Proof.

Let $G = (V, T, S, P)$ be an unrestricted grammar.

We define (where F and E are fresh):

$$
\begin{array}{lll}
w_1 = F & v_1 = FS \Rightarrow & \\
w_2 = \Rightarrow wE & v_2 = E & \\
\vdots \quad x & \vdots \quad y & (x \to y \in P) \\
\quad a & \quad a & (a \in T) \\
\quad A & \quad A & (A \in V) \\
\quad \Rightarrow & \quad \Rightarrow &
\end{array}
$$

This MPCP has a solution $\iff w \in L(G)$.

Contradiction: If the MPCP was decidable, then $w \in L(G)$ was decidable for unrestricted grammars G! □

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example

$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

w_i :

v_i :

Example

$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overset{1}{\overline{F}}$$

$$v_i : \underset{1}{\underline{F\,S\,\Rightarrow}}$$

Example

$$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \ \overset{1}{\overline{F}}\ \overset{3}{\overline{S}}$$

$$v_i: \ \underset{1}{\underline{F\,S \Rightarrow}}\ \underset{3}{\underline{A\,A}}$$

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \overset{1}{\overline{F}}\,\overset{3}{\overline{S}}\,\overset{7}{\overline{\Rightarrow}}$$

$$v_i: \underset{1}{\underline{F\,S\,\Rightarrow}}\,\underset{3}{\underline{A\,A}}\,\underset{7}{\underline{\Rightarrow}}$$

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overset{1}{\overline{F}}\,\overset{3}{\overline{S}}\,\overset{7}{\overline{\Rightarrow}}\,\overset{4}{\overline{A}}$$

$$v_i : \underset{1}{\underline{F\,S\,\Rightarrow}}\,\underset{3}{\underline{A\,A}}\,\underset{7}{\underline{\Rightarrow}}\,\underset{4}{\underline{a\,B}}$$

Example

$$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \underbrace{F}_{1}\,\underbrace{S}_{3}\,\underbrace{\Rightarrow}_{7}\,\underbrace{A}_{4}\,\underbrace{A}_{10}$$

$$v_i: \underbrace{F\,S \Rightarrow}_{1}\,\underbrace{A\,A}_{3}\,\underbrace{\Rightarrow}_{7}\,\underbrace{a\,B}_{4}\,\underbrace{A}_{10}$$

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \quad \overset{1}{\overline{F}}\ \overset{3}{\overline{S}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{4}{\overline{A}}\ \overset{10}{\overline{A}}\ \overset{7}{\overline{\Rightarrow}}$$

$$v_i: \quad \underset{1}{\underline{F\,S \Rightarrow}}\ \underset{3}{\underline{A\,A}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{4}{\underline{a\,B}}\ \underset{10}{\underline{A}}\ \underset{7}{\underline{\Rightarrow}}$$

Example

$$S \to AA \qquad\qquad A \to aB \mid Bb \qquad\qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \quad \overset{1}{\overline{F}}\ \overset{3}{\overline{S}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{4}{\overline{A}}\ \overset{10}{\overline{A}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{8}{\overline{a}}$$

$$v_i: \quad \underset{1}{\underline{F\,S \Rightarrow}}\ \underset{3}{\underline{A\,A}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{4}{\underline{a\,B}}\ \underset{10}{\underline{A}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{8}{\underline{a}}$$

Example

$$S \longrightarrow AA \qquad A \longrightarrow aB \mid Bb \qquad BB \longrightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overset{1}{\overline{F}}\;\overset{3}{\overline{S}}\;\overset{7}{\overline{\Rightarrow}}\;\overset{4}{\overline{A}}\;\overset{10}{\overline{A}}\;\overset{7}{\overline{\Rightarrow}}\;\overset{8}{\overline{a}}\;\overset{11}{\overline{B}}$$

$$v_i : \underset{1}{\underline{F\,S \Rightarrow}}\;\underset{3}{\underline{A\,A}}\;\underset{7}{\underline{\Rightarrow}}\;\underset{4}{\underline{a\,B}}\;\underset{10}{\underline{A}}\;\underset{7}{\underline{\Rightarrow}}\;\underset{8}{\underline{a}}\;\underset{11}{\underline{B}}$$

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overbrace{F}^{1}\,\overbrace{S}^{3}\,\overbrace{\Rightarrow}^{7}\,\overbrace{A}^{4}\,\overbrace{A}^{10}\,\overbrace{\Rightarrow}^{7}\,\overbrace{a}^{8}\,\overbrace{B}^{11}\,\overbrace{A}^{5}$$

$$v_i : \underbrace{F\,S \Rightarrow}_{1}\,\underbrace{A\,A}_{3}\,\underbrace{\Rightarrow}_{7}\,\underbrace{a\,B}_{4}\,\underbrace{A}_{10}\,\underbrace{\Rightarrow}_{7}\,\underbrace{a}_{8}\,\underbrace{B}_{11}\,\underbrace{B\,b}_{5}$$

Example

$$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

w_i :

1	3	7	4	10	7	8	11	5	7
F	S	$\Rightarrow$	A	A	$\Rightarrow$	a	B	A	$\Rightarrow$

v_i :

$FS \Rightarrow$	AA	$\Rightarrow$	aB	A	$\Rightarrow$	a	B	Bb	$\Rightarrow$
1	3	7	4	10	7	8	11	5	7

Example

$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i: \quad \overset{1}{F}\ \overset{3}{S}\ \overset{7}{\Rightarrow}\ \overset{4}{A}\ \overset{10}{A}\ \overset{7}{\Rightarrow}\ \overset{8}{a}\ \overset{11}{B}\ \overset{5}{A}\ \overset{7}{\Rightarrow}\ \overset{8}{a}$$

$$v_i: \quad \underset{1}{FS\Rightarrow}\ \underset{3}{AA}\ \underset{7}{\Rightarrow}\ \underset{4}{aB}\ \underset{10}{A}\ \underset{7}{\Rightarrow}\ \underset{8}{a}\ \underset{11}{B}\ \underset{5}{Bb}\ \underset{7}{\Rightarrow}\ \underset{8}{a}$$

Example

$$S \to AA \qquad A \to aB \mid Bb \qquad BB \to aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overset{1}{\overline{F}}\ \overset{3}{\overline{S}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{4}{\overline{A}}\ \overset{10}{\overline{A}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{8}{\overline{a}}\ \overset{11}{\overline{B}}\ \overset{5}{\overline{A}}\ \overset{7}{\overline{\Rightarrow}}\ \overset{8}{\overline{a}}\ \overset{6}{\overline{BB}}$$

$$v_i : \underset{1}{\underline{FS \Rightarrow}}\ \underset{3}{\underline{AA}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{4}{\underline{aB}}\ \underset{10}{\underline{A}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{8}{\underline{a}}\ \underset{11}{\underline{B}}\ \underset{5}{\underline{Bb}}\ \underset{7}{\underline{\Rightarrow}}\ \underset{8}{\underline{a}}\ \underset{6}{\underline{aa}}$$

Example

$$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

$$w_i : \overset{1}{\overline{F}}\,\overset{3}{\overline{S}}\,\overset{7}{\overline{\Rightarrow}}\,\overset{4}{\overline{A}}\,\overset{10}{\overline{A}}\,\overset{7}{\overline{\Rightarrow}}\,\overset{8}{\overline{a}}\,\overset{11}{\overline{B}}\,\overset{5}{\overline{A}}\,\overset{7}{\overline{\Rightarrow}}\,\overset{8}{\overline{a}}\,\overset{6}{\overline{BB}}\,\overset{9}{\overline{b}}$$

$$v_i : \underset{1}{\underline{FS\Rightarrow}}\,\underset{3}{\underline{AA}}\,\underset{7}{\underline{\Rightarrow}}\,\underset{4}{\underline{aB}}\,\underset{10}{\underline{A}}\,\underset{7}{\underline{\Rightarrow}}\,\underset{8}{\underline{a}}\,\underset{11}{\underline{B}}\,\underset{5}{\underline{Bb}}\,\underset{7}{\underline{\Rightarrow}}\,\underset{8}{\underline{a}}\,\underset{6}{\underline{aa}}\,\underset{9}{\underline{b}}$$

Example

$$S \rightarrow AA \qquad A \rightarrow aB \mid Bb \qquad BB \rightarrow aa$$

This grammar with $w = aaab$ translates to the following MPCP:

i	w_i	v_i
1	F	$FS \Rightarrow$
2	$\Rightarrow aaabE$	E
3	S	AA
4	A	aB
5	A	Bb
6	BB	aa

i	w_i	v_i
7	$\Rightarrow$	$\Rightarrow$
8	a	a
9	b	b
10	A	A
11	B	B
12	S	S

Example derivation: $S \Rightarrow AA \Rightarrow aBA \Rightarrow aBBb \Rightarrow aaab$.

w_i :

1	3	7	4	10	7	8	11	5	7	8	6	9	2
F	S	$\Rightarrow$	A	A	$\Rightarrow$	a	B	A	$\Rightarrow$	a	BB	b	$\Rightarrow aaabE$

v_i :

1	3	7	4	10	7	8	11	5	7	8	6	9	2
$FS \Rightarrow$	AA	$\Rightarrow$	aB	A	$\Rightarrow$	a	B	Bb	$\Rightarrow$	a	aa	b	E

Post Correspondence Problem

Theorem

The PCP is undecidable.

Post Correspondence Problem

Theorem

The PCP is undecidable.

Proof.

Given an MPCP X: $(w_1, v_1), \ldots, (w_n, v_n)$ where

$$w_i = a_{i1} \cdots a_{im_i} \quad \text{and} \quad v_i = b_{i1} \cdots b_{ir_i} \quad (\text{with } m_i + r_i > 0)$$

Post Correspondence Problem

Theorem

The PCP is undecidable.

Proof.

Given an MPCP X: $(w_1, v_1), \ldots, (w_n, v_n)$ where

$$w_i = a_{i1} \cdots a_{im_i} \quad \text{and} \quad v_i = b_{i1} \cdots b_{ir_i} \quad (\text{with } m_i + r_i > 0)$$

We define PCP X' $(y_0, z_0), \ldots, (y_{n+1}, z_{n+1})$ by:

$$y_0 = @\$y_1 \qquad y_i = a_{i1}\$a_{i2}\$\cdots a_{im_i}\$ \qquad y_{n+1} = \#$$

$$z_0 = @z_1 \qquad z_i = \$b_{i1}\$b_{i2}\cdots\$b_{ir_i} \qquad z_{n+1} = \$\#$$

for $1 \leq i \leq n$. The letters $@$, $\$$ and # are fresh.

Post Correspondence Problem

Theorem

The PCP is undecidable.

Proof.

Given an MPCP X: $(w_1, v_1), \ldots, (w_n, v_n)$ where

$$w_i = a_{i1} \cdots a_{im_i} \quad \text{and} \quad v_i = b_{i1} \cdots b_{ir_i} \quad (\text{with } m_i + r_i > 0)$$

We define PCP X' $(y_0, z_0), \ldots, (y_{n+1}, z_{n+1})$ by:

$$y_0 = @\$y_1 \qquad y_i = a_{i1}\$a_{i2}\$\cdots a_{im_i}\$ \qquad y_{n+1} = \#$$

$$z_0 = @z_1 \qquad z_i = \$b_{i1}\$b_{i2}\cdots\$b_{ir_i} \qquad z_{n+1} = \$\#$$

for $1 \leq i \leq n$. The letters @, \$ and # are fresh.

Every PCP X' solution must start with (y_0, z_0):

$$y_0 y_j \cdots y_k y_{n+1} = z_0 z_j \cdots z_k z_{n+1}$$

Post Correspondence Problem

Theorem

The PCP is undecidable.

Proof.

Given an MPCP X: $(w_1, v_1), \ldots, (w_n, v_n)$ where

$$w_i = a_{i1} \cdots a_{im_i} \quad \text{and} \quad v_i = b_{i1} \cdots b_{ir_i} \quad (\text{with } m_i + r_i > 0)$$

We define PCP X' $(y_0, z_0), \ldots, (y_{n+1}, z_{n+1})$ by:

$$y_0 = @\$y_1 \qquad y_i = a_{i1}\$a_{i2}\$ \cdots a_{im_i}\$ \qquad y_{n+1} = \#$$

$$z_0 = @z_1 \qquad z_i = \$b_{i1}\$b_{i2} \cdots \$b_{ir_i} \qquad z_{n+1} = \$\#$$

for $1 \leq i \leq n$. The letters $@$, $\$$ and # are fresh.

Every PCP X' solution must start with (y_0, z_0):

$$y_0 y_j \cdots y_k y_{n+1} = z_0 z_j \cdots z_k z_{n+1}$$

Solution exists $\iff$ $w_1 w_j \cdots w_k = v_1 v_j \cdots v_k$ is a solution of X.

Post Correspondence Problem

Theorem

The PCP is undecidable.

Proof.

Given an MPCP X: $(w_1, v_1), \ldots, (w_n, v_n)$ where

$$w_i = a_{i1} \cdots a_{im_i} \quad \text{and} \quad v_i = b_{i1} \cdots b_{ir_i} \quad (\text{with } m_i + r_i > 0)$$

We define PCP X' $(y_0, z_0), \ldots, (y_{n+1}, z_{n+1})$ by:

$$y_0 = @\$y_1 \qquad y_i = a_{i1}\$a_{i2}\$ \cdots a_{im_i}\$ \qquad y_{n+1} = \#$$

$$z_0 = @z_1 \qquad z_i = \$b_{i1}\$b_{i2} \cdots \$b_{ir_i} \qquad z_{n+1} = \$\#$$

for $1 \leq i \leq n$. The letters $@$, $\$$ and # are fresh.

Every PCP X' solution must start with (y_0, z_0):

$$y_0 y_j \cdots y_k y_{n+1} = z_0 z_j \cdots z_k z_{n+1}$$

Solution exists $\iff$ $w_1 w_j \cdots w_k = v_1 v_j \cdots v_k$ is a solution of X.

As the MPCP is undecidable, so must be the PCP. □

Example

Consider the following instance of the MPCP:

$$w_1 = 11 \qquad w_2 = 1$$
$$v_1 = 1 \qquad v_2 = 11$$

It reduces to the following PCP problem:

$$y_0 = @\$1\$1\$ \qquad y_1 = 1\$1\$ \qquad y_2 = 1\$ \qquad y_3 = \#$$
$$z_0 = @\$1 \qquad z_1 = \$1 \qquad z_2 = \$1\$1 \qquad z_3 = \$\#$$

Example

Consider the following instance of the MPCP:

$$w_1 = 11 \qquad w_2 = 1$$
$$v_1 = 1 \qquad v_2 = 11$$

It reduces to the following PCP problem:

$$y_0 = @\$1\$1\$ \qquad y_1 = 1\$1\$ \qquad y_2 = 1\$ \qquad y_3 = \#$$
$$z_0 = @\$1 \qquad z_1 = \$1 \qquad z_2 = \$1\$1 \qquad z_3 = \$\#$$

Example solution MPCP:

$$w_1 w_2 = 111 = v_1 v_2$$

Corresponding solution PCP:

$$y_0 y_2 y_3 = @\$1\$1\$1\$\# = z_0 z_2 z_3$$

Example

Consider the following instance of the MPCP:

$$w_1 = 11 \qquad w_2 = 1$$
$$v_1 = 1 \qquad v_2 = 11$$

It reduces to the following PCP problem:

$$y_0 = @\$1\$1\$ \qquad y_1 = 1\$1\$ \qquad y_2 = 1\$ \qquad y_3 = \#$$
$$z_0 = @\$1 \qquad z_1 = \$1 \qquad z_2 = \$1\$1 \qquad z_3 = \$\#$$

Example solution MPCP:

$$w_1 w_2 = 111 = v_1 v_2$$

Corresponding solution PCP:

$$y_0 y_2 y_3 = @\$1\$1\$1\$\# = z_0 z_2 z_3$$

In general: the original MPCP instance has a solution
$\iff$ the resulting PCP instance has a solution

Undecidable Properties of Context-Free Languages

Undecidable properties of context-free languages:

- empty intersection,
- ambiguity,
- palindromes,
- equality,
- ...

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$ **?** for context-free languages L_1, L_2 is undecidable.

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$ **?** for context-free languages L_1, L_2 is undecidable.

Proof.

We reduce the PCP to the above problem.

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$ **?** for context-free languages L_1, L_2 is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$? for context-free languages L_1, L_2 is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define two context-free grammars G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

for $1 \leq i \leq n$. Here #, $\langle$ and $\rangle$ are fresh symbols.

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$? for context-free languages L_1, L_2 is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define two context-free grammars G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

for $1 \le i \le n$. Here #, $\langle$ and $\rangle$ are fresh symbols. Then

$$L(G_1) = \{w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle \mid 1 \le j, \ldots, k \le n\}$$
$$L(G_2) = \{v_\ell \cdots v_m \# \langle m \rangle \cdots \langle \ell \rangle \mid 1 \le \ell, \ldots, m \le n\}$$

Empty Intersection of Context-Free Languages

Theorem

The question $L_1 \cap L_2 = \emptyset$ **?** for context-free languages L_1, L_2 is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define two context-free grammars G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \to v_i\, S_2 \langle i \rangle \mid v_i\, \# \langle i \rangle$$

for $1 \leq i \leq n$. Here #, $\langle$ and $\rangle$ are fresh symbols. Then

$$L(G_1) = \{w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle \mid 1 \leq j, \ldots, k \leq n\}$$
$$L(G_2) = \{v_\ell \cdots v_m \# \langle m \rangle \cdots \langle \ell \rangle \mid 1 \leq \ell, \ldots, m \leq n\}$$

$L(G_1) \cap L(G_2) = \emptyset \iff$ the PCP X has no solution. □

Ambiguity of Context-Free Grammars

Theorem

Ambiguity of context-free grammars is undecidable.

Ambiguity of Context-Free Grammars

Theorem

Ambiguity of context-free grammars is undecidable.

Proof.

We reduce the PCP to the above problem.

Ambiguity of Context-Free Grammars

Theorem

Ambiguity of context-free grammars is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

Ambiguity of Context-Free Grammars

Theorem

Ambiguity of context-free grammars is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define a context-free grammar G:

$$S \to S_1 \mid S_2 \qquad\qquad S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

for $1 \leq i \leq n$. Here #, $\langle$ and $\rangle$ are fresh symbols.

Ambiguity of Context-Free Grammars

Theorem

Ambiguity of context-free grammars is undecidable.

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define a context-free grammar G:

$$S \to S_1 \mid S_2 \qquad\qquad S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

for $1 \leq i \leq n$. Here #, $\langle$ and $\rangle$ are fresh symbols.

Then G is ambiguous $\iff$ the PCP X has a solution. □

Palindromes in Context-Free Languages

Theorem

It is undecidable whether a context-free languages contains a palindrome (a word $w = w^R$).

Palindromes in Context-Free Languages

Theorem

It is undecidable whether a context-free languages contains a palindrome (a word $w = w^R$).

Proof.

We reduce the PCP to the above problem.

Palindromes in Context-Free Languages

Theorem

It is undecidable whether a context-free languages contains a palindrome (a word $w = w^R$).

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

Palindromes in Context-Free Languages

Theorem

It is undecidable whether a context-free languages contains a palindrome (a word $w = w^R$).

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define a context-free grammar G:

$$S \to w_i\, S\, v_i^R \mid w_i \,\#\, v_i^R$$

for $1 \leq i \leq n$. Here # is a fresh symbol.

Palindromes in Context-Free Languages

Theorem

It is undecidable whether a context-free languages contains a palindrome (a word $w = w^R$).

Proof.

We reduce the PCP to the above problem.

Given a PCP instance X: $(w_1, v_1), \ldots, (w_n, v_n)$.

We define a context-free grammar G:

$$S \to w_i \, S \, v_i^R \mid w_i \, \# \, v_i^R$$

for $1 \leq i \leq n$. Here # is a fresh symbol.

$L(G)$ contains a palindrome $\iff$ PCP X has a solution. □

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$.

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \rightarrow w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \rightarrow v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before.

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \rightarrow w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

$$S_2 \rightarrow v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before. Then

$$\text{PCP } X \text{ has no solution} \iff L(G_1) \cap L(G_2) = \emptyset$$

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before. Then

$$\text{PCP } X \text{ has no solution} \iff L(G_1) \cap L(G_2) = \emptyset$$

$$\iff \overline{L(G_1) \cap L(G_2)} = \overline{\emptyset}$$

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before. Then

$$\begin{aligned} \text{PCP } X \text{ has no solution} &\iff L(G_1) \cap L(G_2) = \emptyset \\ &\iff \overline{L(G_1) \cap L(G_2)} = \overline{\emptyset} \\ &\iff \overline{L(G_1)} \cup \overline{L(G_2)} = \Sigma^* \end{aligned}$$

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before. Then

$$\begin{aligned} \text{PCP } X \text{ has no solution} &\iff L(G_1) \cap L(G_2) = \emptyset \\ &\iff \overline{L(G_1) \cap L(G_2)} = \overline{\emptyset} \\ &\iff \overline{L(G_1)} \cup \overline{L(G_2)} = \Sigma^* \end{aligned}$$

It suffices to show that $\overline{L(G_1)} \cup \overline{L(G_2)}$ is context-free.

Equality of Context-Free Languages

Theorem

The question $L = \Sigma^*$? (and hence $L_1 = L_2$?) for context-free languages L (L_1, L_2) is undecidable.

Proof

Given a PCP X: $(w_1, v_1), \ldots, (w_n, v_n)$. Define G_1 and G_2:

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$
$$S_2 \to v_i S_2 \langle i \rangle \mid v_i \# \langle i \rangle$$

as before. Then

$$\begin{aligned} \text{PCP } X \text{ has no solution} &\iff L(G_1) \cap L(G_2) = \emptyset \\ &\iff \overline{L(G_1) \cap L(G_2)} = \overline{\emptyset} \\ &\iff \overline{L(G_1)} \cup \overline{L(G_2)} = \Sigma^* \end{aligned}$$

It suffices to show that $\overline{L(G_1)} \cup \overline{L(G_2)}$ is context-free.

It suffices that $\overline{L(G_1)}$ is context-free ($\overline{L(G_2)}$ is analogous).

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \rightarrow w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \leq j, \ldots, k \leq n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{\#\} \cdot \{\langle 1 \rangle, \ldots, \langle n \rangle\}^+.$$

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \leq j, \ldots, k \leq n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{\#\} \cdot \{\langle 1 \rangle, \ldots, \langle n \rangle\}^+.$$

We have $L(G_1) \subseteq L_S$, so

$\overline{L(G_1)} = \Sigma^* \setminus L(G_1) = (L_S \cup \overline{L_S}) \setminus L(G_1) = (L_S \setminus L(G_1)) \cup \overline{L_S}$

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \rightarrow w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \leq j, \ldots, k \leq n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{ \# \} \cdot \{ \langle 1 \rangle, \ldots, \langle n \rangle \}^+ .$$

We have $L(G_1) \subseteq L_S$, so

$$\overline{L(G_1)} = \Sigma^* \setminus L(G_1) = (L_S \cup \overline{L_S}) \setminus L(G_1) = (L_S \setminus L(G_1)) \cup \overline{L_S}$$

As L_S is regular, also $\overline{L_S}$ is regular (and context-free).

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \rightarrow w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \leq j, \ldots, k \leq n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{\#\} \cdot \{\langle 1 \rangle, \ldots, \langle n \rangle\}^+.$$

We have $L(G_1) \subseteq L_S$, so

$\overline{L(G_1)} = \Sigma^* \setminus L(G_1) = (L_S \cup \overline{L_S}) \setminus L(G_1) = (L_S \setminus L(G_1)) \cup \overline{L_S}$

As L_S is regular, also $\overline{L_S}$ is regular (and context-free).

So it suffices to show that $L_S \setminus L(G_1)$ is context-free.

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \leq j, \ldots, k \leq n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{\#\} \cdot \{\langle 1 \rangle, \ldots, \langle n \rangle\}^+.$$

We have $L(G_1) \subseteq L_S$, so

$$\overline{L(G_1)} = \Sigma^* \setminus L(G_1) = (L_S \cup \overline{L_S}) \setminus L(G_1) = (L_S \setminus L(G_1)) \cup \overline{L_S}$$

As L_S is regular, also $\overline{L_S}$ is regular (and context-free).

So it suffices to show that $L_S \setminus L(G_1)$ is context-free.

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{ w \# \langle k \rangle \cdots \langle j \rangle \mid w \neq w_j \cdots w_k \}$$

Equality of Context-Free Languages (2)

Proof continued

$$S_1 \to w_i S_1 \langle i \rangle \mid w_i \# \langle i \rangle$$

The words in $L(G_1)$ are of the form

$w_j \cdots w_k \# \langle k \rangle \cdots \langle j \rangle$ for non-empty indices $1 \le j, \ldots, k \le n$

All these words are of the shape

$$L_S = \Sigma^* \cdot \{\#\} \cdot \{ \langle 1 \rangle, \ldots, \langle n \rangle \}^+.$$

We have $L(G_1) \subseteq L_S$, so

$\overline{L(G_1)} = \Sigma^* \setminus L(G_1) = (L_S \cup \overline{L_S}) \setminus L(G_1) = (L_S \setminus L(G_1)) \cup \overline{L_S}$

As L_S is regular, also $\overline{L_S}$ is regular (and context-free).

So it suffices to show that $L_S \setminus L(G_1)$ is context-free.

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{ w \# \langle k \rangle \cdots \langle j \rangle \mid w \ne w_j \cdots w_k \}$$

We distinguish three cases…

Equality of Context-Free Languages (3)

Proof continued

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{\, w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; w \neq w_j \cdots w_k \,\}$$

We distinguish three cases:

Equality of Context-Free Languages (3)

Proof continued

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{\, w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; w \neq w_j \cdots w_k \,\}$$

We distinguish three cases:

$$L_S \setminus L(G_1) = L_{\text{smaller}} \cup L_{\text{larger}} \cup L_{\text{equal}}$$

where

$$\begin{aligned}
L_{\text{smaller}} &= \{ w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| < |w_j \cdots w_k| \} \\
L_{\text{larger}} &= \{ w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| > |w_j \cdots w_k| \} \\
L_{\text{equal}} &= \{ w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| = |w_j \cdots w_k| \;\&\; w \neq w_j \ldots w_k \}
\end{aligned}$$

Equality of Context-Free Languages (3)

Proof continued

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{\, w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; w \neq w_j \cdots w_k \,\}$$

We distinguish three cases:

$$L_S \setminus L(G_1) = L_{\text{smaller}} \cup L_{\text{larger}} \cup L_{\text{equal}}$$

where

$$\begin{aligned} L_{\text{smaller}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| < |w_j \cdots w_k|\} \\ L_{\text{larger}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| > |w_j \cdots w_k|\} \\ L_{\text{equal}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| = |w_j \cdots w_k| \;\&\; w \neq w_j \ldots w_k\} \end{aligned}$$

Each of these languages is context-free, thus $L_S \setminus L(G_1)$ is.

Equality of Context-Free Languages (3)

Proof continued

The words in $L_S \setminus L(G_1)$ are of the form:

$$L_S \setminus L(G_1) = \{\, w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; w \neq w_j \cdots w_k \,\}$$

We distinguish three cases:

$$L_S \setminus L(G_1) = L_{\text{smaller}} \cup L_{\text{larger}} \cup L_{\text{equal}}$$

where

$$\begin{aligned} L_{\text{smaller}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| < |w_j \cdots w_k|\} \\ L_{\text{larger}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| > |w_j \cdots w_k|\} \\ L_{\text{equal}} &= \{w \,\#\, \langle k \rangle \cdots \langle j \rangle \;\mid\; |w| = |w_j \cdots w_k| \;\&\; w \neq w_j \ldots w_k\} \end{aligned}$$

Each of these languages is context-free, thus $L_S \setminus L(G_1)$ is.

Exercise

Give context-free grammars for L_{smaller}, L_{larger} and L_{equal}.

Semidecidability

Recall that a decision $P \subseteq \Sigma^*$ is called

- **decidable** if the P is recursive,
- **semidecidable** if the P is recursively enumerable.

Semidecidability

Recall that a decision $P \subseteq \Sigma^*$ is called

- **decidable** if the P is recursive,
- **semidecidable** if the P is recursively enumerable.

Examples of (undecidable but) semidecidable problems:

- halting problem,
- Post correspondence problem,
- non-empty intersection of context-free languages,
- ambiguity of context-free grammars.

Semidecidability

Recall that a decision $P \subseteq \Sigma^*$ is called

- **decidable** if the P is recursive,
- **semidecidable** if the P is recursively enumerable.

Examples of (undecidable but) semidecidable problems:

- halting problem,
- Post correspondence problem,
- non-empty intersection of context-free languages,
- ambiguity of context-free grammars.

There exist algorithms for these problems that always halt if the answer is yes, but may or may not halt if the answer is no.

More Undecidable Problems

Derivability of a formula ϕ in **predicate logic** is undecidable**.**

(Logic and Modelling)

More Undecidable Problems

Derivability of a formula ϕ in **predicate logic** is undecidable.

(Logic and Modelling)

In 1900 **David Hilbert** (1862-1941) formulated 23 scientific problems.

More Undecidable Problems

Derivability of a formula ϕ in **predicate logic** is undecidable.

(Logic and Modelling)

In 1900 **David Hilbert** (1862-1941) formulated 23 scientific problems. Among them the following:

Diophantine equations consist of polynomials with one or more variables and coefficients in $\mathbb{Z}$. For example:

$$3x^2y - 7y^2z^3 - 18 = 0$$
$$-7y^2 + 8z^3 = 0$$

More Undecidable Problems

Derivability of a formula ϕ in **predicate logic** is undecidable.

(Logic and Modelling)

In 1900 **David Hilbert** (1862-1941) formulated 23 scientific problems. Among them the following:

Diophantine equations consist of polynomials with one or more variables and coefficients in $\mathbb{Z}$. For example:

$$3x^2y - 7y^2z^3 - 18 = 0$$
$$-7y^2 + 8z^3 = 0$$

Hilbert's 10th problem: Give an algorithm to decide whether a system of Diophantine equations has a solution in $\mathbb{Z}$.

More Undecidable Problems

Derivability of a formula ϕ in **predicate logic** is undecidable.

(Logic and Modelling)

In 1900 **David Hilbert** (1862-1941) formulated 23 scientific problems. Among them the following:

Diophantine equations consist of polynomials with one or more variables and coefficients in $\mathbb{Z}$. For example:

$$3x^2y - 7y^2z^3 - 18 = 0$$
$$-7y^2 + 8z^3 = 0$$

Hilbert's 10th problem: Give an algorithm to decide whether a system of Diophantine equations has a solution in $\mathbb{Z}$.

In 1970 **Yuri Matiyasevich** proved that this is undecidable.