Undecidable problems for Recursively enumerable languages

continued...

class 21

Take a recursively enumerable language L

Decision problems:

- L is empty?
- L is finite?
- L contains two different strings of the same length?

All these problems are undecidable

Theorem:

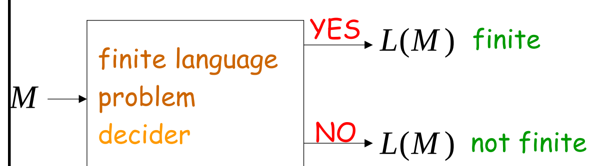
For a recursively enumerable language L it is undecidable to determine whether L is finite

Proof:

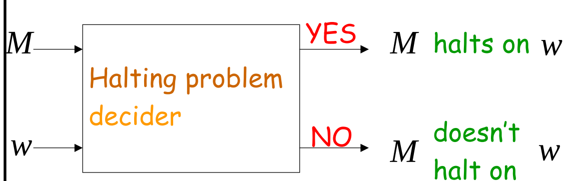
We will reduce the halting problem to this problem

Let M be the TM with $L(M) = L$

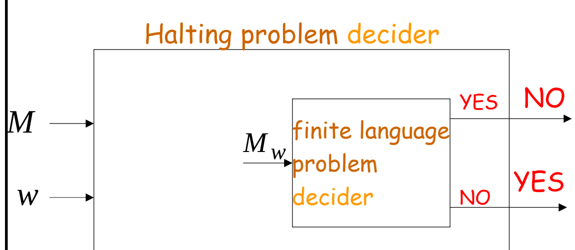
Suppose we have a decider for the finite language problem:



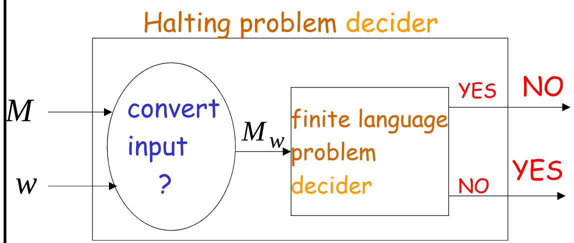
We will build a decider for the halting problem:



We want to reduce the halting problem to the finite language problem



We need to convert one problem instance to the other problem instance



7

Construct machine M_w:

On arbitrary input string s

Initially, simulates M on input w

If M enters a halt state, accept s (Σ^* inifinite language)

Otherwise, reject s ($\varnothing$ finite language)

8

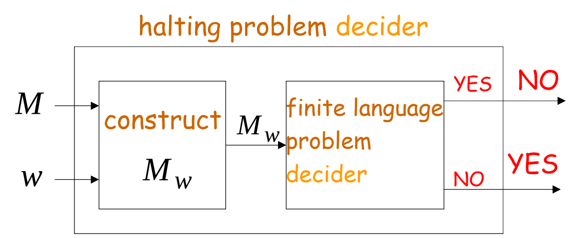
M halts on w

if and only if

$L(M_w)$ is infinite

$$L(M_w) = \Sigma^*$$

9



10

Take a recursively enumerable language L

Decision problems:

- L is empty?
- L is finite?
- L contains two different strings of the same length?

All these problems are undecidable

11

Theorem:

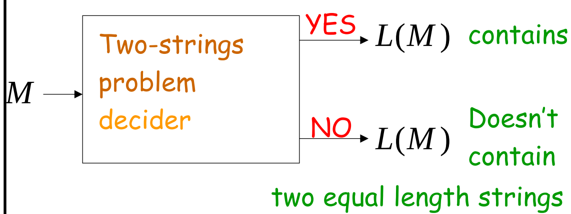
For a recursively enumerable language L it is undecidable to determine whether L contains two different strings of same length

Proof: We will reduce the halting problem to this problem

12

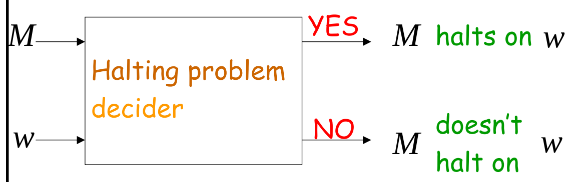
Let M be the TM with $L(M)=L$

Suppose we have the decider for the two-strings problem:



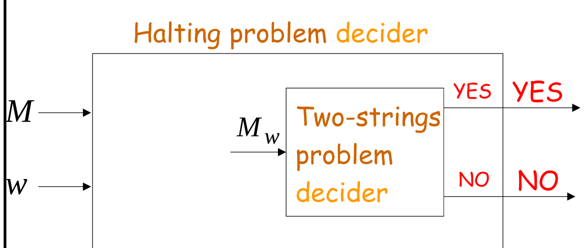
13

We will build a decider for the halting problem:



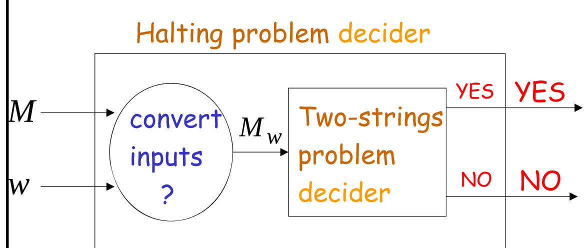
14

We want to reduce the halting problem to the empty language problem



15

We need to convert one problem instance to the other problem instance



16

Construct machine M_w :

On arbitrary input string s

Initially, simulate M on input w

When M enters a halt state, accept if $s=a$ or $s=b$

(two equal length strings $L(M_w)=\{a,b\}$)

Otherwise, reject s ($L(M_w)=\varnothing$)

17

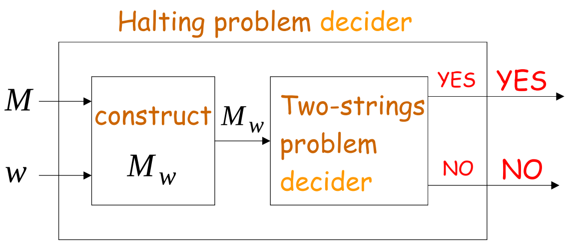
M halts on w

if and only if

M_w accepts two equal length strings

M_w accepts a and b

18



19

Rice's Theorem

20

Definition:

Non-trivial properties of recursively enumerable languages:

any property possessed by some (not all) recursively enumerable languages

21

Some non-trivial properties of recursively enumerable languages:

- L is empty
- L is finite
- L contains two different strings of the same length

22

Rice's Theorem:

Any non-trivial property of a recursively enumerable language is undecidable

23

The Post Correspondence Problem

24

Some undecidable problems for context-free languages:

- Is $L(G_1) \cap L(G_2) = \varnothing$?

  G_1, G_2 are context-free grammars

- Is context-free grammar G ambiguous?

25

We need a tool to prove that the previous problems for context-free languages are undecidable:

The Post Correspondence Problem

26

The Post Correspondence Problem

Input: Two sequences of n strings

$$A = w_1, w_2, \ldots, w_n$$

$$B = v_1, v_2, \ldots, v_n$$

27

There is a **P**ost **C**orrespondence Solution if there is a sequence $i, j, \ldots, k$ such that:

PC-solution: $w_i w_j \cdots w_k = v_i v_j \cdots v_k$

Indeces may be repeated or ommited

28

Example:

	w_1	w_2	w_3
A:	100	11	111
	v_1	v_2	v_3
B:	001	111	11

PC-solution: 2,1,3 $w_2 w_1 w_3 = v_2 v_1 v_3$

11100111

29

Example:

	w_1	w_2	w_3
A:	00	001	1000
	v_1	v_2	v_3
B:	0	11	011

There is no solution

Because total length of strings from B is smaller than total length of strings from A

30

The Modified Post Correspondence Problem

Inputs: $A = w_1, w_2, \ldots, w_n$

$B = v_1, v_2, \ldots, v_n$

MPC-solution: $1, i, j, \ldots, k$

$$w_1 w_i w_j \cdots w_k = v_1 v_i v_j \cdots v_k$$

31

Example:

	w_1	w_2	w_3
$A:$	11	111	100

	v_1	v_2	v_3
$B:$	111	11	001

MPC-solution: 1,3,2 $\quad w_1 w_3 w_2 = v_1 v_3 v_2$

11100111

32

We will show:

1. The MPC problem is undecidable
   (by reducing the membership to MPC)

2. The PC problem is undecidable
   (by reducing MPC to PC)

33

Theorem: The MPC problem is undecidable

Proof: We will reduce the membership problem to the MPC problem

34

Membership problem

Input: recursive language L
string w

Question: $w \in L$?

Undecidable

35

Membership problem

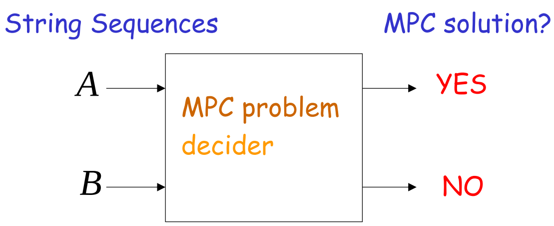
Input: unrestricted grammar G
string w

Question: $w \in L(G)$?

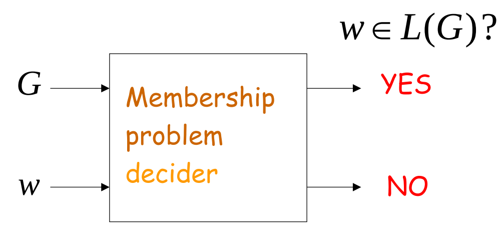
Undecidable

36

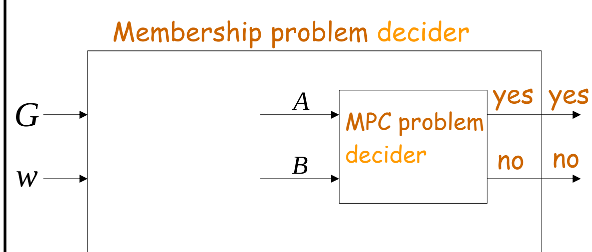
Suppose we have a decider for
the MPC problem



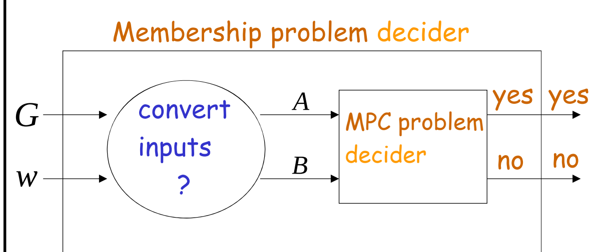
We will build a decider for
the membership problem



The reduction of the membership problem
to the MPC problem:



We need to convert the input instance of
one problem to the other



A	B	Grammar G
$FS \Rightarrow$	F	S : start variable F : special symbol
a	a	For every symbol a
V	V	For every variable V

A	B	Grammar G
E	$\Rightarrow wE$	string w E : special symbol
y	x	For every production $x \rightarrow y$
$\Rightarrow$	$\Rightarrow$	

Example:

Grammar G : $S \to aABb \mid Bbb$

$Bb \to C$

$AC \to aac$

String $w = aaac$

A		B	
$w_1:$	$FS \Rightarrow$	$v_1:$	F
$w_2:$	a	$v_2:$	a
	b		b
	c		c
$\vdots$	A	$\vdots$	A
	B		B
	C		C
$w_8:$	S	$v_8:$	S

A		B	
$w_9:$	$E \Rightarrow$	$v_9:$	$aaacE$
	$aABb$		S
$\vdots$	Bbb	$\vdots$	S
	C		Bb
	aac		AC
$w_{14}:$	$\Rightarrow$	$v_{14}:$	$\Rightarrow$

Grammar G : $S \to aABb \mid Bbb$

$Bb \to C$

$AC \to aac$

$$aaac \in L(G)$$

$$S \Rightarrow aABb \Rightarrow aAC \Rightarrow aaac$$

$$S \Rightarrow aABb$$

A

$$\overbrace{\underbrace{F}_{v_1}\ \underbrace{S}_{v_{10}}\ \Rightarrow}^{w_1}\ \overbrace{a\ A\ B\ b}^{w_{10}}$$

B

$$S \Rightarrow aABb \Rightarrow aAC$$

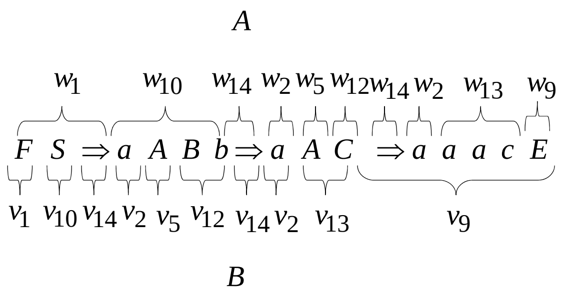
A

$$\overbrace{F\ S\ \Rightarrow}^{w_1}\ \overbrace{a\ A\ B\ b}^{w_{10}}\ \overbrace{\Rightarrow}^{w_{14}}\ \overbrace{a}^{w_2}\ \overbrace{A}^{w_5}\ \overbrace{C}^{w_{12}}$$

$$\underbrace{F}_{v_1}\ \underbrace{S}_{v_{10}}\ \underbrace{\Rightarrow}_{v_{14}}\ \underbrace{a}_{v_2}\ \underbrace{A}_{v_5}\ \underbrace{B\ b}_{v_{12}}\ \Rightarrow a\ A\ C$$

B

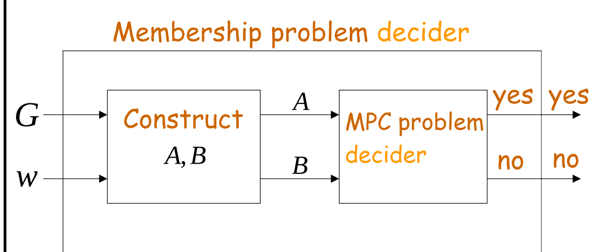
$S \Rightarrow aABb \Rightarrow aAC \Rightarrow aaac$



(A, B) has an MPC-solution

if and only if

$w \in L(G)$



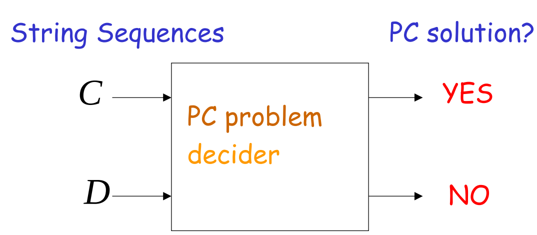
Since the membership problem is undecidable,
The MPC problem is uncedecidable

END OF PROOF

Theorem: The PC problem is undecidable

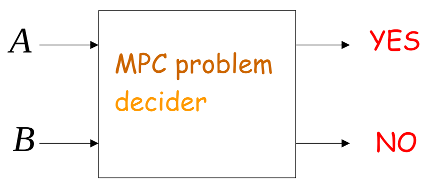
Proof: We will reduce the MPC problem to the PC problem

Suppose we have a decider for the PC problem

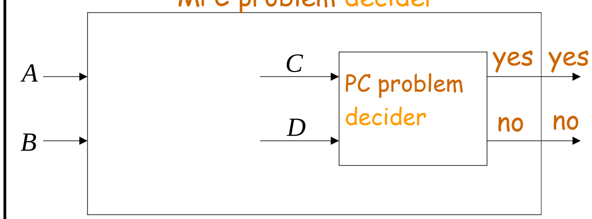


We will build a decider for the MPC problem

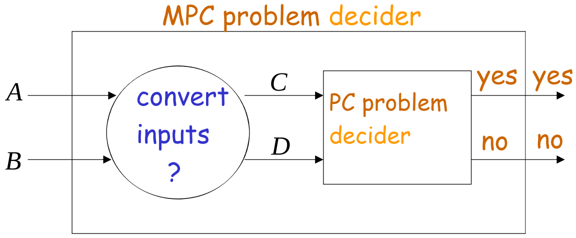
String Sequences MPC solution?



The reduction of the MPC problem to the PC problem:



We need to convert the input instance of one problem to the other



A, B : input to the MPC problem

$$A = w_1, w_2, \ldots, w_n$$

$$B = v_1, v_2, \ldots, v_n$$

We construct new sequences C, D

$$C = w'_0, w'_1, \ldots, w'_n, w'_{n+1}$$

$$D = v'_0, v'_1, \ldots, v'_n, v'_{n+1}$$

$$A = w_1, w_2, \ldots, w_n$$

$$B = v_1, v_2, \ldots, v_n$$

A $\qquad w_i = abcad$

C $\qquad w'_i = a*b*c*a*d*$

We insert a special symbol between any two symbols

$B \qquad v_i = abcad$

$D \qquad v_i' = *a*b*c*a*d$

61

Special Cases

C	D
$w_0' = *w_1$	$v_0' = v_1'$
$w_{n+1}' = \Diamond$	$v_{n+1}' = *\Diamond$

62

C, D has a PC solution

if and only if

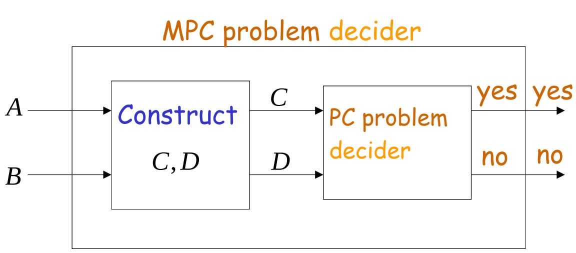
A, B has an MPC solution

63

PC-solution $\quad w_0' \cdots w_k' w_{n+1}' = v_0' \cdots w_k' v_{n+1}'$

MPC-solution $\quad w_1 \cdots w_k = v_1 \cdots v_k$

64



65

Since the MPC problem is undecidable,
The PC problem is undecidable

END OF PROOF

66

Some undecidable problems for context-free languages:

- Is $L(G_1) \cap L(G_2) = \varnothing$?

  G_1, G_2 are context-free grammars

- Is context-free grammar G ambiguous?

We reduce the PC problem to these problems

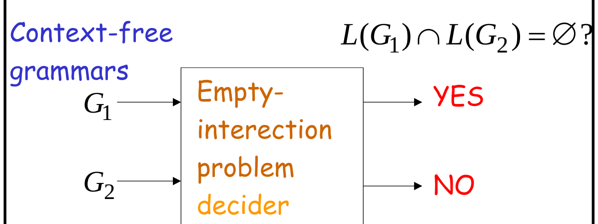
67

Theorem: Let G_1, G_2 be context-free grammars. It is undecidable to determine if $L(G_1) \cap L(G_2) = \varnothing$

Proof: Rdeduce the PC problem to this problem

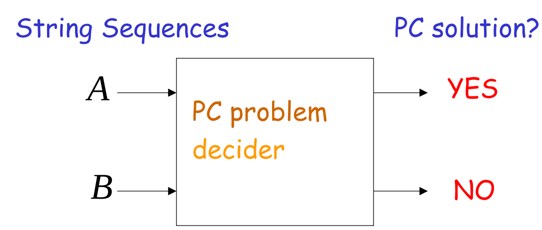
68

Suppose we have a decider for the empty-intersection problem



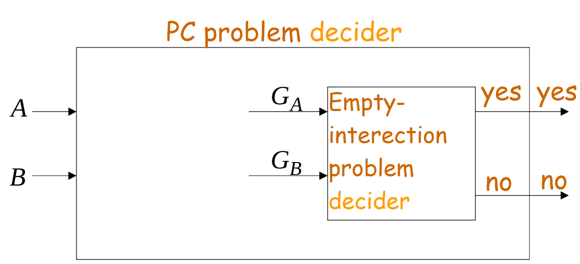
69

We will build a decider for the PC problem



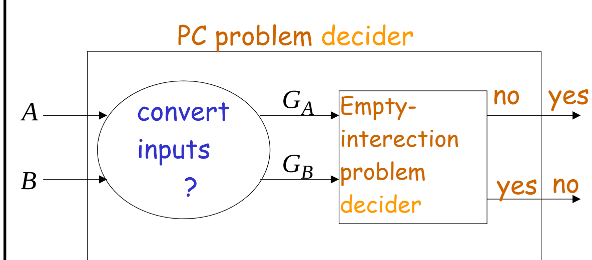
70

The reduction of the PC problem to the empty-intersection problem:



71

We need to convert the input instance of one problem to the other



72

A, B : input to the PC problem

$$A = w_1, w_2, \ldots, w_n$$

$$B = v_1, v_2, \ldots, v_n$$

$$A = w_1, w_2, \ldots, w_n$$

$$B = v_1, v_2, \ldots, v_n$$

Introduce new unique symbols:

$$a_1, a_2, \ldots, a_n$$

$$a_1, a_2, \ldots, a_n$$

$$A = w_1, w_2, \ldots, w_n$$

$$L_A = \{s: \quad s = w_i w_j \cdots w_k a_k \cdots a_j a_i\}$$

Context-free grammar G_A:

$$S_A \to w_i S_A a_i \mid w_i a_i$$

$$a_1, a_2, \ldots, a_n$$

$$B = v_1, v_2, \ldots, v_n$$

$$L_B = \{s: \quad s = v_i v_j \cdots v_k a_k \cdots a_j a_i\}$$

Context-free grammar G_B:

$$S_B \to v_i S_B a_i \mid v_i a_i$$

(A, B) has a PC solution

if and only if

$$L(G_A) \cap L(G_B) \neq \emptyset$$

$$L(G_1) \cap L(G_2) \neq \emptyset$$

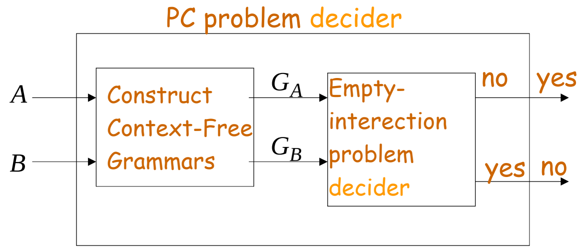
$$s = w_i w_j \cdots w_k a_k \cdots a_j a_i$$

$$s = v_i v_j \cdots v_k a_k \cdots a_j a_i$$

Because $a_1, a_2, \ldots, a_n$ are unique

There is a PC solution:

$$s = w_i w_j \cdots w_k a_k \cdots a_j a_i$$

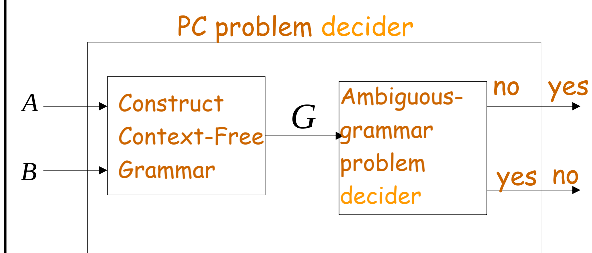


Since PC is undecidable,
the empty-intersection problem is undecidable

END OF PROOF

Theorem: For a context-free grammar G,
it is undecidable to determine
if G is ambiguous

Proof: Reduce the PC problem
to this problem



S_A start variable of G_A

S_B start variable of G_B

S start variable of G

$$S \rightarrow S_A \mid S_B$$

(A, B) has a PC solution

if and only if

$$L(G_A) \cap L(G_B) \neq \varnothing$$

if and only if

G is ambiguous