The Post Correspondence Problem

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

The Post Correspondence Problem

The Post Correspondence Problem

Input: Two sets of n strings

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

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

There is a Post Correspondence Solution if there is a sequence $i, j, \ldots, k$ such that:

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

Indices may be repeated or omitted

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

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

The Modified Post Correspondence Problem

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

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

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

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

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 $w_1w_3w_2 = v_1v_3v_2$

11100111

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)

Theorem: The MPC problem is undecidable

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

Membership problem

Input: Turing machine M

string w

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

Undecidable

Membership problem

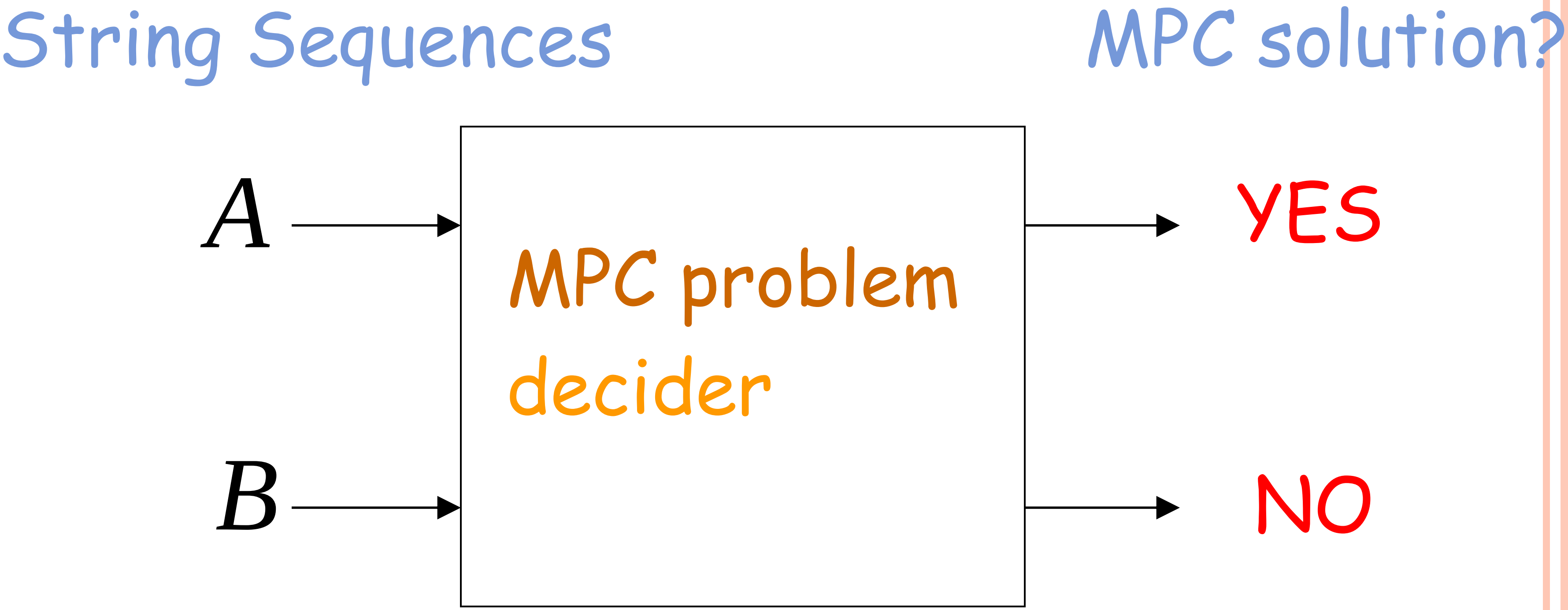
Input: unrestricted grammar G

string w

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

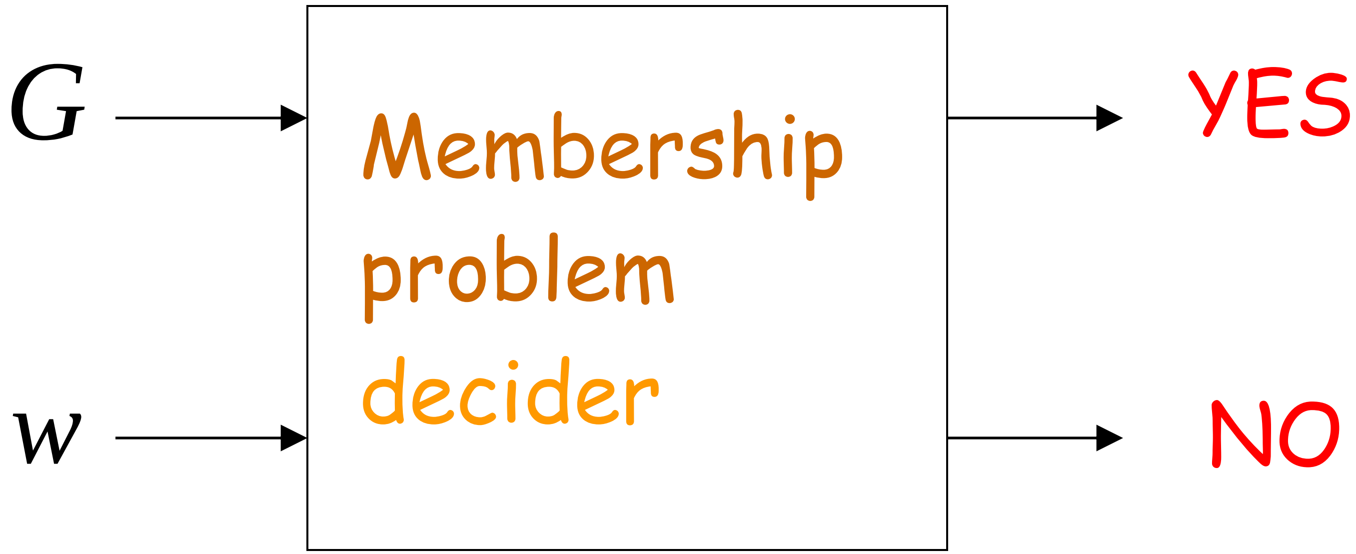
<u>Undecidable</u>

Suppose we have a decider for the MPC problem

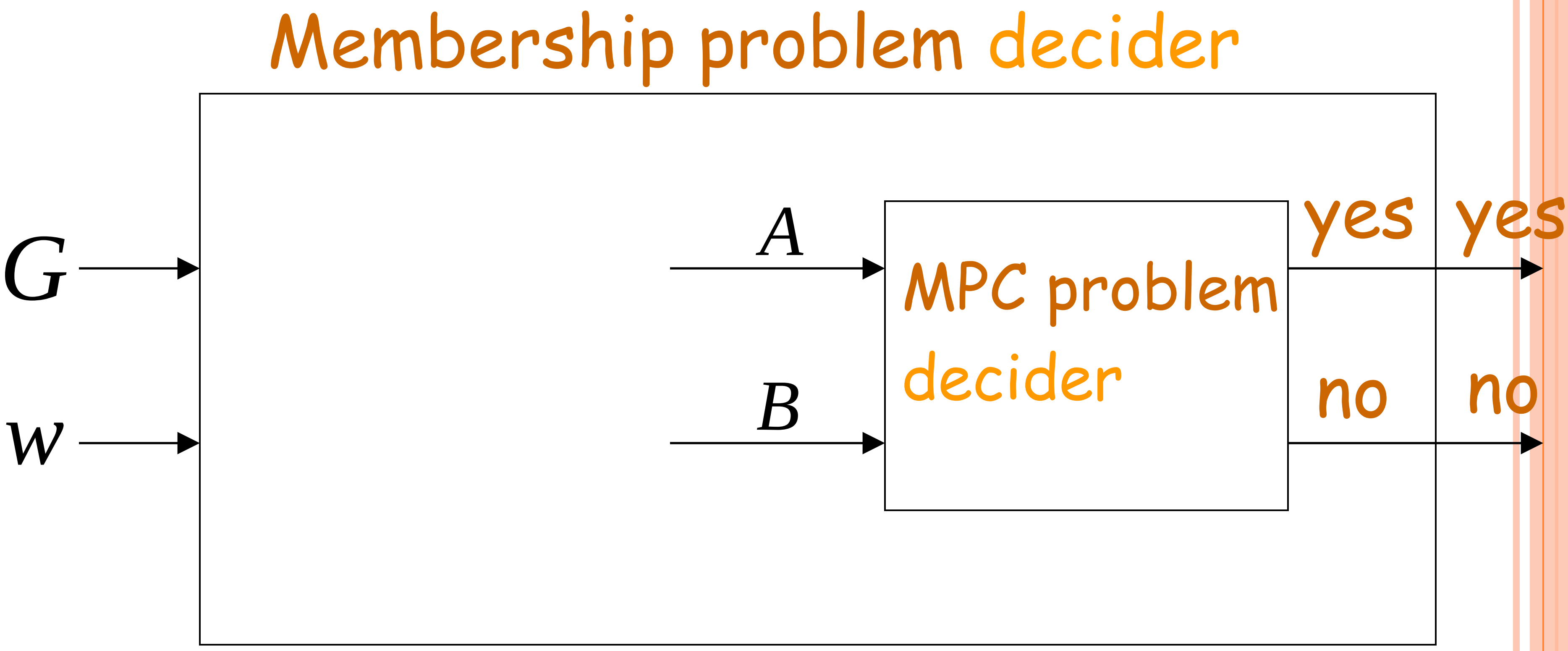


We will build a decider for the membership problem

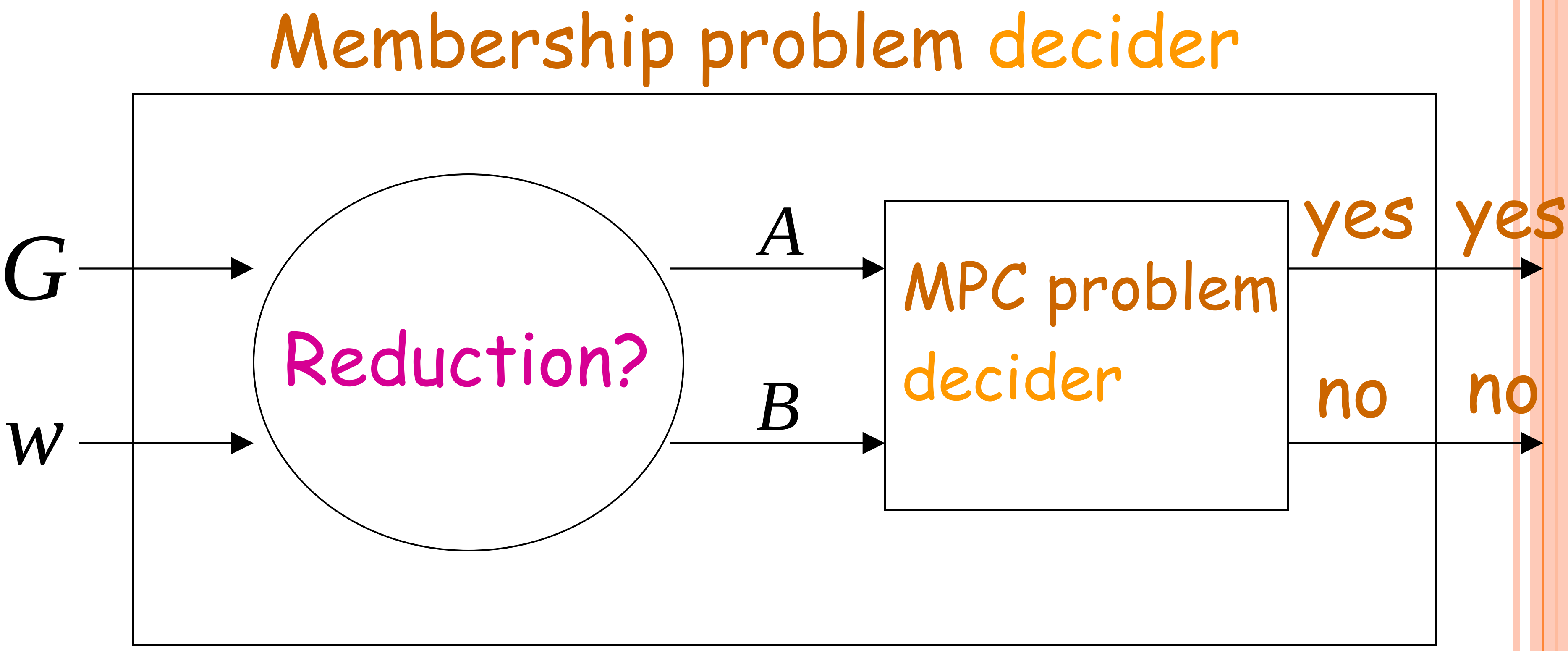
$$w \in L(G)?$$



The reduction of the membership problem to the MPC problem:



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



Reduction:

Convert grammar G and string w

to sets of strings A and B

Such that:

G generates w $\Longleftrightarrow$ There is an MPC solution for A, B

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
w_3 :	b	v_3 :	b
	c		c
$\vdots$	A	$\vdots$	A
	B		B
	C		C
w_8 :	S	v_8 :	S

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

Grammar G :

$$S \to aABb \mid Bbb$$

$$Bb \to C$$

$$AC \to aac$$

$aaac \in L(G)$:

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

$S \rightarrow aABb \mid Bbb$

$Bb \rightarrow C$

$AC \rightarrow aac$

Derivation: S

$A:$ $\overbrace{F\ \ S \Rightarrow}^{w_1}$

$B:$ $\underbrace{F}_{v_1}$

$$S \to aABb \mid Bbb$$
$$Bb \to C$$
$$AC \to aac$$

Derivation:

$$S \Rightarrow aABb$$

A:

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

B: $v_1\ v_{10}$

$$S \to aABb \mid Bbb$$
$$Bb \to C$$
$$AC \to aac$$

Derivation:

$$S \Rightarrow aABb \Rightarrow aAC$$

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

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

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

A: $w_1 \quad w_{10} \quad w_{14} \quad w_2 \quad w_5 \quad w_{12}$

B: $v_1 \quad v_{10} \quad v_{14} \quad v_2 \quad v_5 \quad v_{12}$

Derivation:

$$S \rightarrow aABb \mid Bbb$$
$$Bb \rightarrow C$$
$$AC \rightarrow aac$$

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

A: $w_1 \quad w_{10} \quad w_{14} \quad w_2 \quad w_5 \quad w_{12} \quad w_{14} \quad w_2 \quad w_{13}$

$$F \; S \Rightarrow a \; A \; B \; b \Rightarrow a \; A \; C \Rightarrow a \; a \; a \; c \; E$$

B: $v_1 \quad v_{10} \quad v_{14} \quad v_2 \quad v_5 \quad v_{12} \quad v_{14} \quad v_2 \quad v_{13}$

Derivation:

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

$S \rightarrow aABb \mid Bbb$

$Bb \rightarrow C$

$AC \rightarrow aac$

A: $w_1 \quad w_{10} \quad w_{14} \quad w_2 \quad w_5 \quad w_{12} \quad w_{14} \quad w_2 \quad w_{13} \quad w_9$

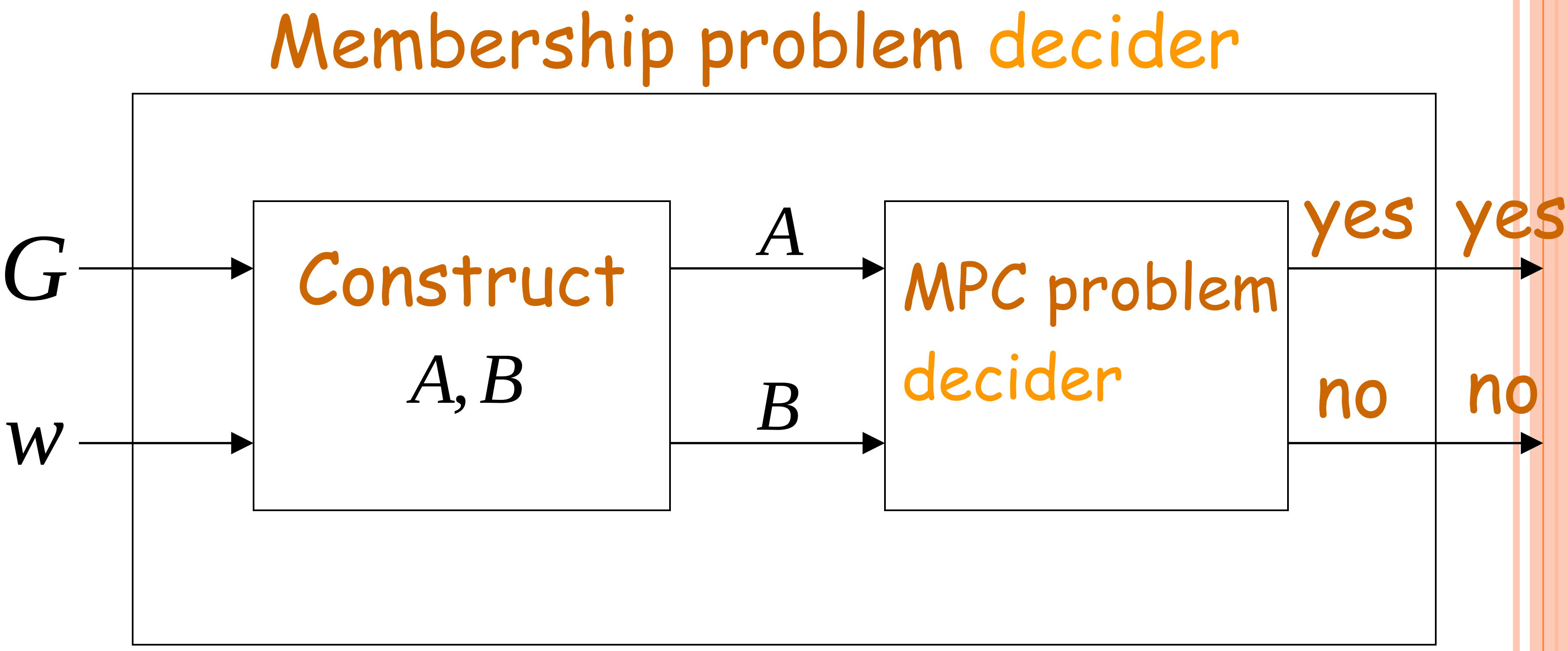
$$F \; S \Rightarrow a \; A \; B \; b \Rightarrow a \; A \; C \Rightarrow a \; a \; a \; c \; E$$

B: $v_1 \quad v_{10} \quad v_{14} \quad v_2 \quad v_5 \quad v_{12} \quad v_{14} \quad v_2 \quad v_{13} \quad v_9$

(A, B) has an MPC-solution

if and only if

$w \in L(G)$


Membership problem decider
G
w
Construct
A, B
A
B
MPC problem
decider
yes
yes
no
no

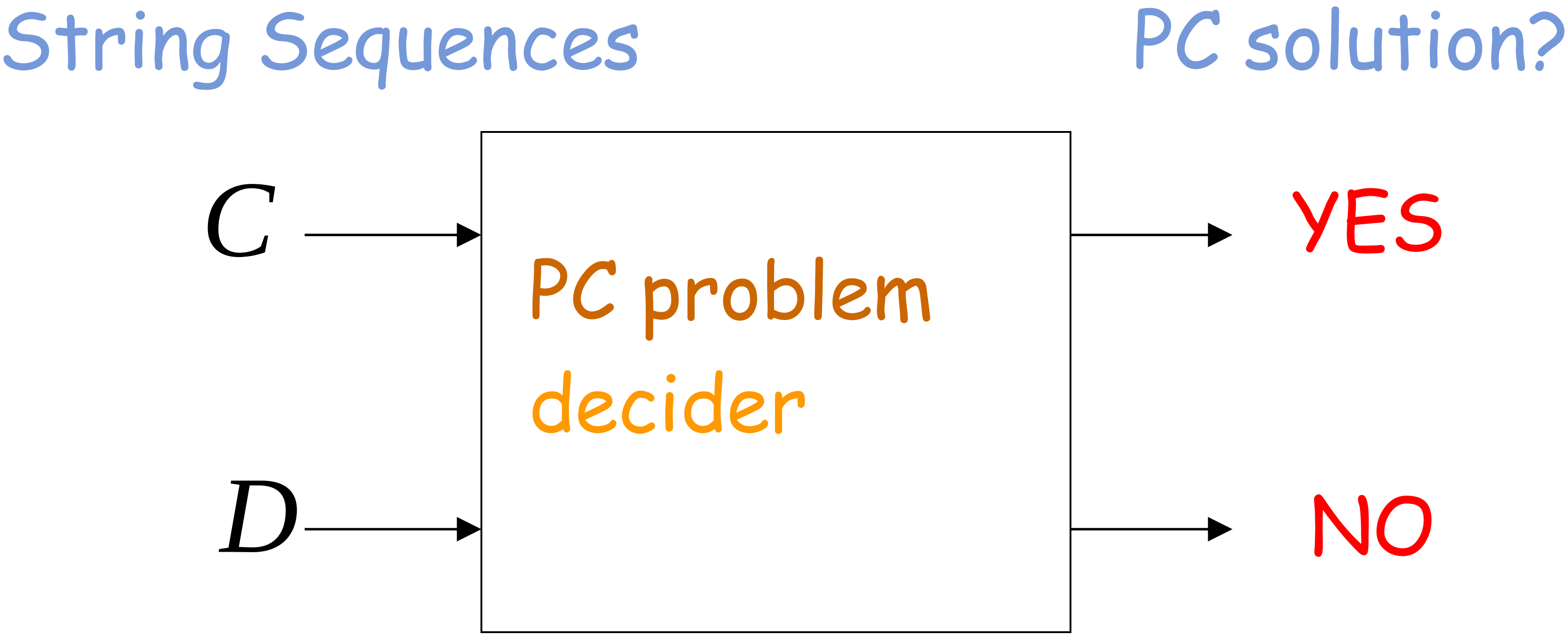
Since the membership problem is undecidable,
The MPC problem is undecidable

END OF PROOF

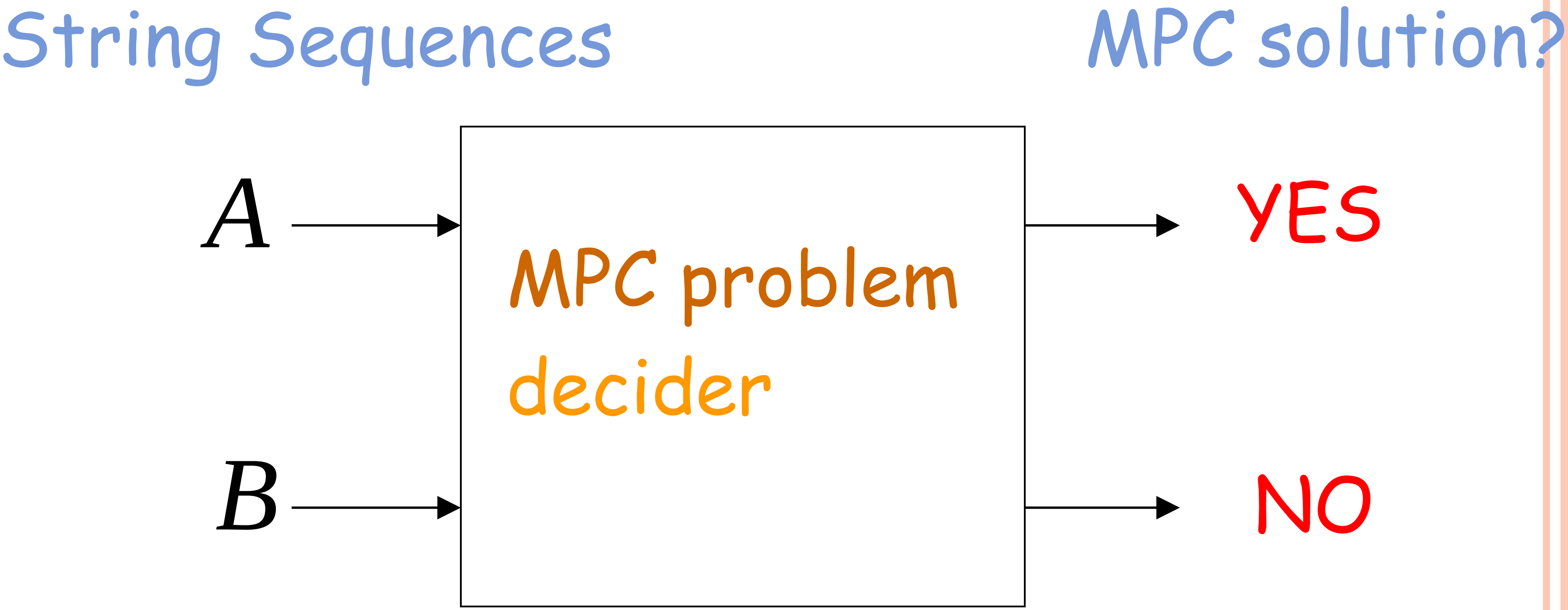
Theorem: The PC problem is undecidable

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

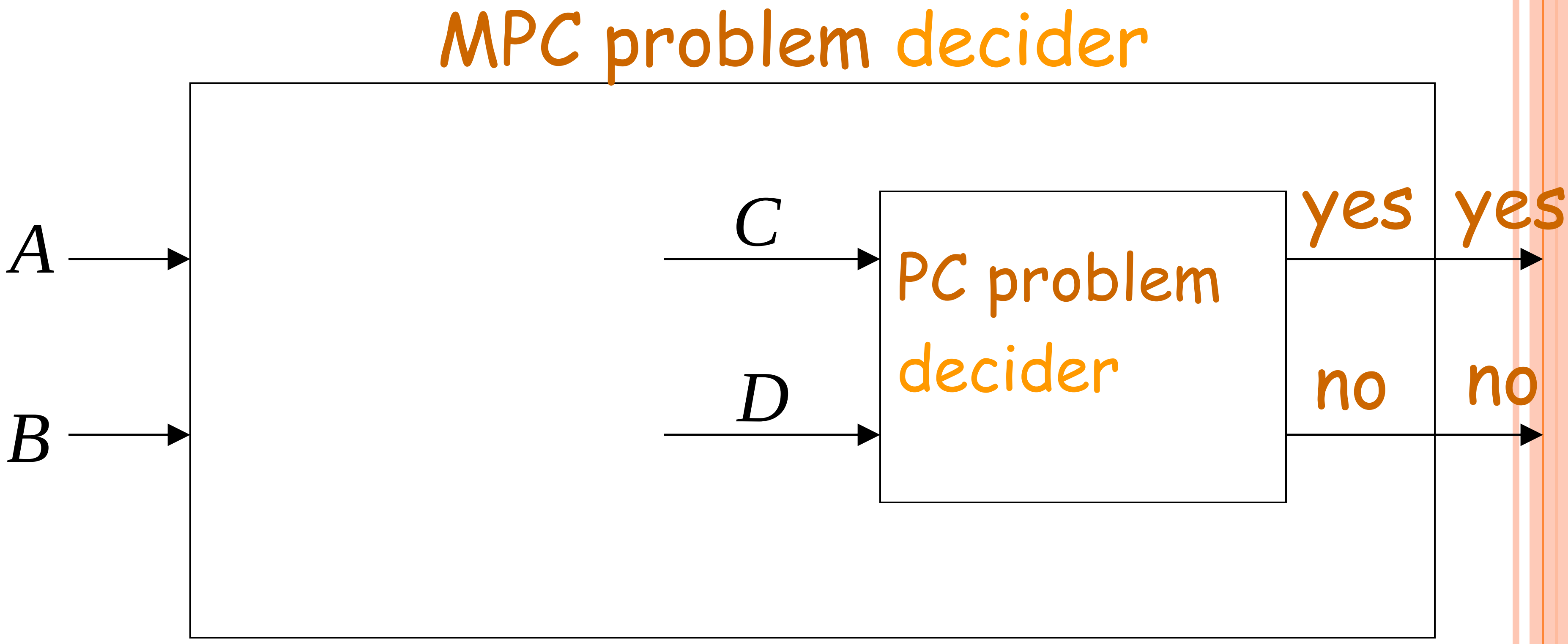
Suppose we have a decider for the PC problem



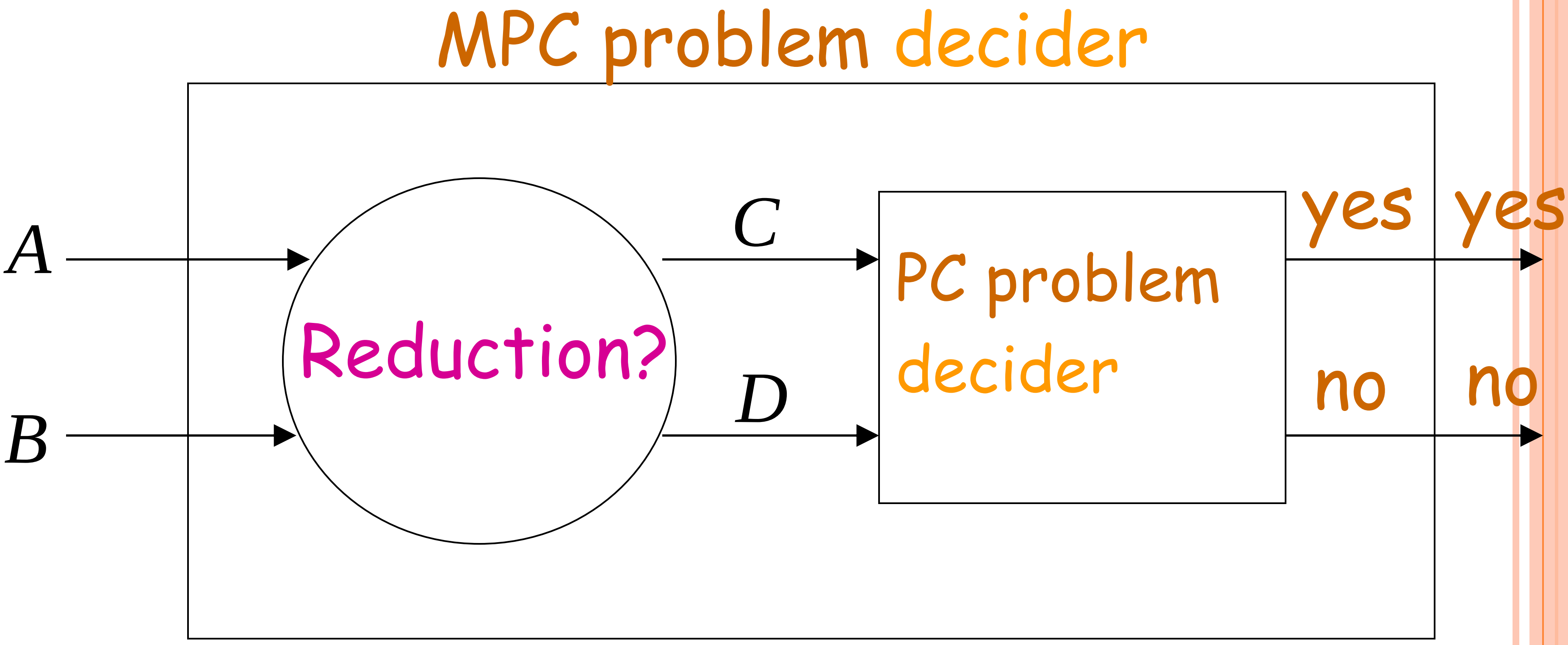
We will build a decider for the MPC problem



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

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

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

Translated to

C, D : input to the PC problem

$$C = w_1', \ldots, w_n', w_{n+1}'$$

$$D = v_1', \ldots, v_n', v_{n+1}'$$

A

$w_i = \sigma_1 \sigma_2 \cdots \sigma_k$ ⟹ C: $w'_i = \sigma_1 * \sigma_2 * \cdots \sigma_k *$

For each i

replace $w'_1 = * w'_1$

$w'_{n+1} = \Diamond$

B

$v_i = \pi_1 \pi_2 \cdots \pi_k$ ⟹ D: $v'_i = * \pi_1 * \pi_2 * \cdots * \pi_k$

For each i

$v'_{n+1} = * \Diamond$

PC-solution

$$\overset{C}{w_1' w_i' \cdots w_k' w_{n+1}'} = \overset{D}{v_1' v_i' \cdots w_k' v_{n+1}'}$$

Has to start with
These strings

C PC-solution D

$$w'_1 w'_i \cdots w'_k w'_{n+1} = v'_1 v'_i \cdots w'_k v'_{n+1}$$

A B

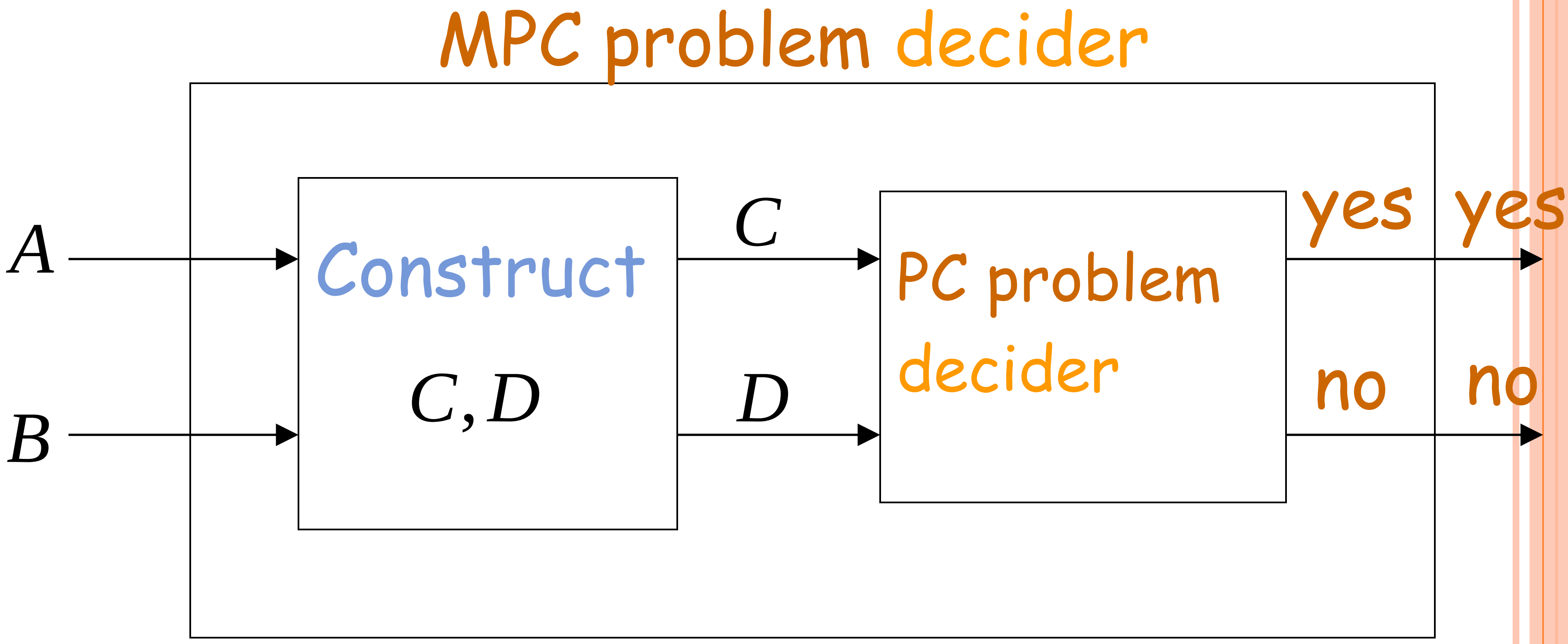
$$w_1 w_i \cdots w_k = v_1 v_i \cdots v_k$$

MPC-solution

C, D has a PC solution

if and only if

A, B has an MPC solution


MPC problem decider
A
B
Construct
C, D
C
D
PC problem
decider
yes
no
yes
no

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

END OF PROOF

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

Theorem: Let G_1, G_2 be context-free grammars. It is undecidable to determine if

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

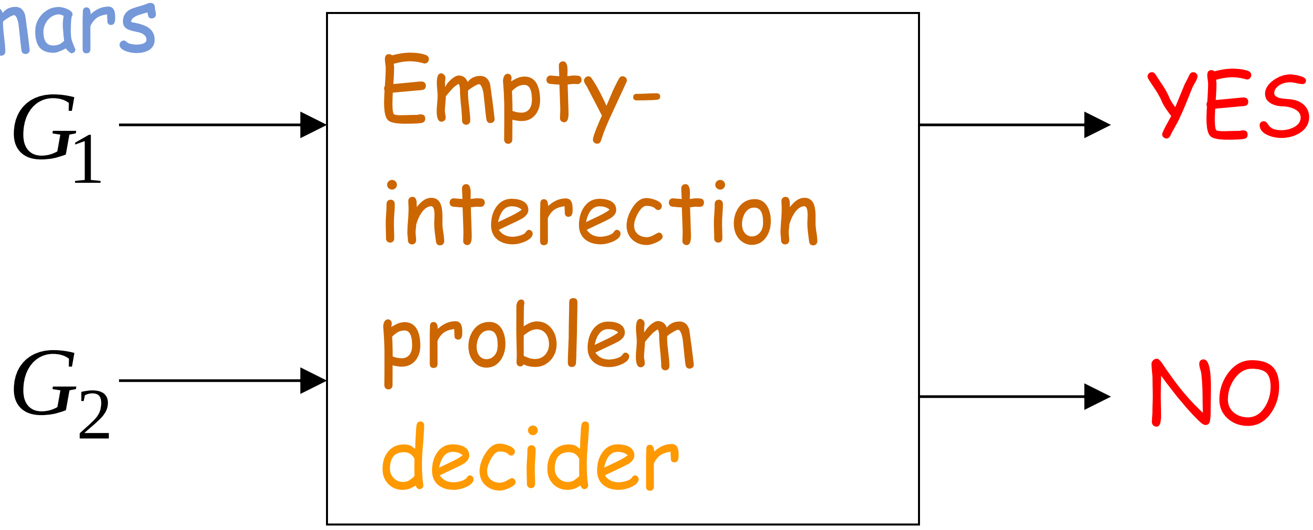
(intersection problem)

Proof: Reduce the PC problem to this problem

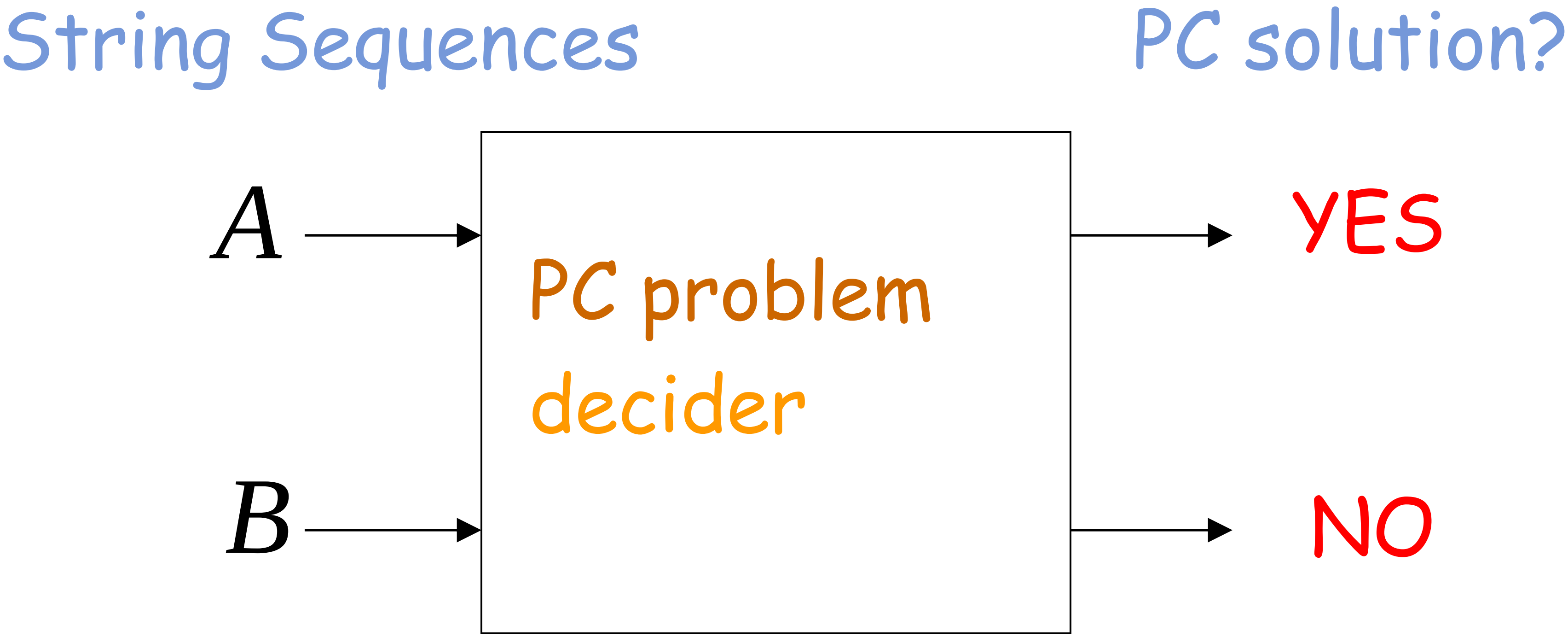
Suppose we have a decider for the intersection problem

Context-free grammars

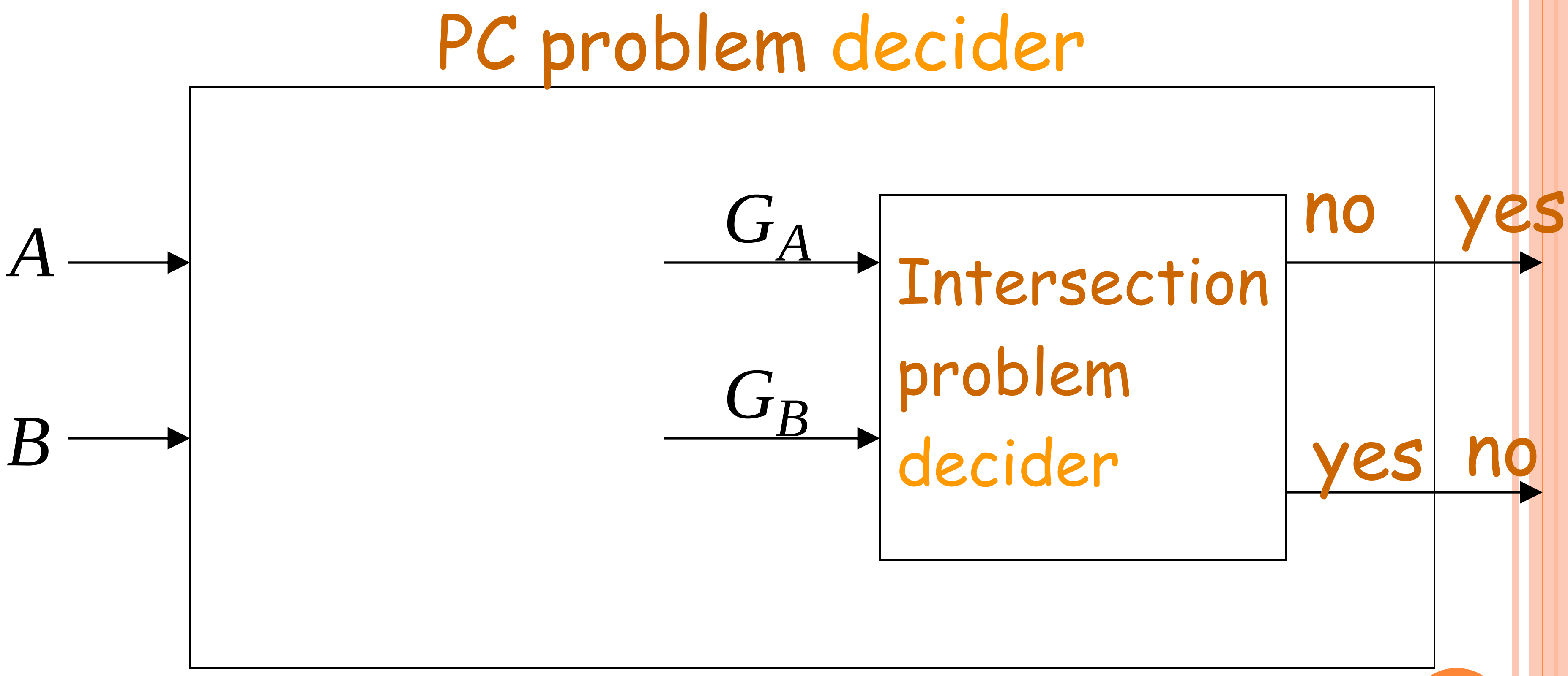
$$L(G_1) \cap L(G_2) = \emptyset ?$$



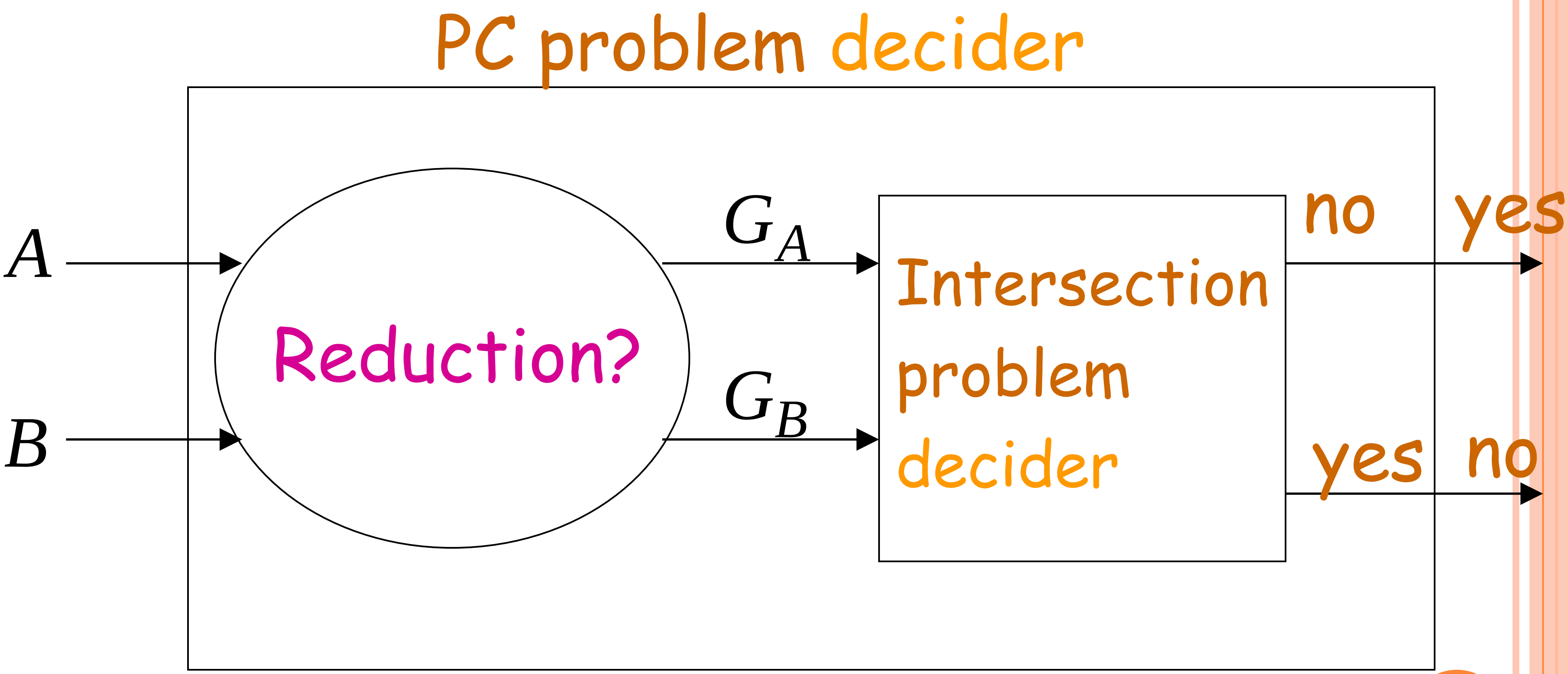
We will build a decider for the PC problem



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



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



Introduce new unique symbols: $a_1, a_2, \ldots, a_n$

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

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

Context-free grammar G_A: $S_A \rightarrow w_i S_A a_i \mid w_i a_i$

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

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

Context-free grammar G_B: $S_B \rightarrow v_i S_B a_i \mid v_i a_i$

(A, B) has a PC solution

if and only if

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

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

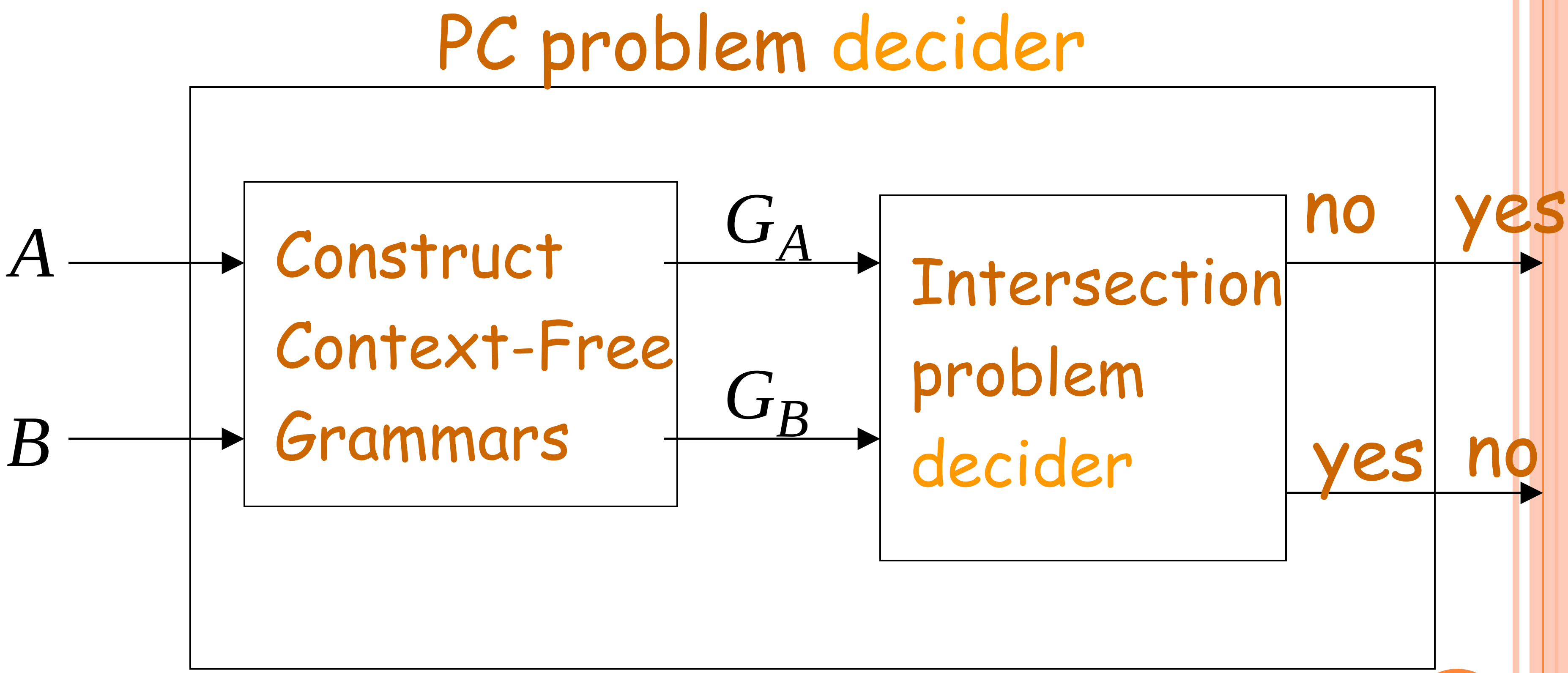
$$s = w_i w_j \cdots w_k a_k \cdots a_j a_i$$

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

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

There is a PC solution:

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


PC problem decider
A
B
Construct
Context-Free
Grammars
G_A
G_B
Intersection
problem
decider
no
yes
yes
no

Since PC is undecidable,
the Intersection problem is undecidable

END OF PROOF

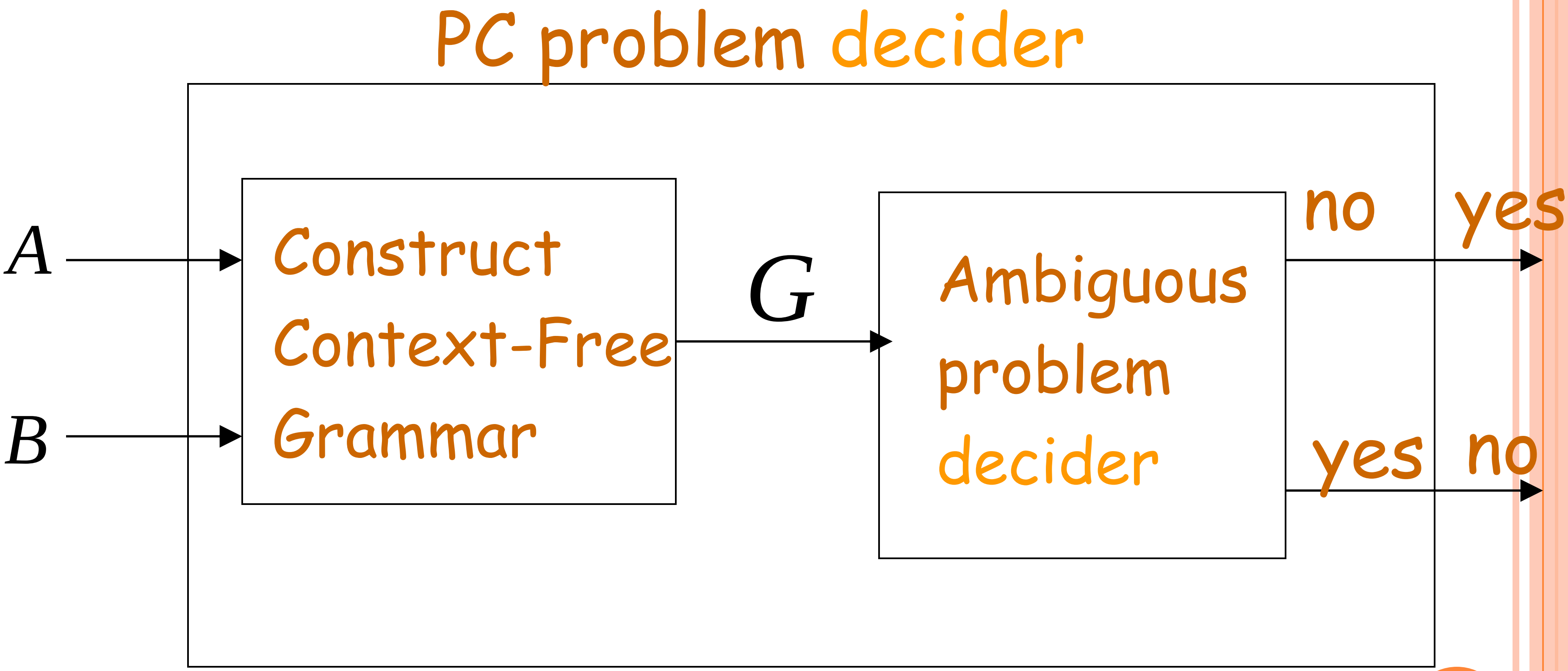
Theorem: For a context-free grammar G,

it is undecidable to determine

if G is ambiguous

Proof: Reduce the PC problem

to this problem


PC problem decider
A
B
Construct
Context-Free
Grammar
G
Ambiguous
problem
decider
no
yes
yes
no

S_A start variable of G_A

S_B start variable of G_B

S start variable of G

$$S \to S_A \mid S_B$$

(A, B) has a PC solution

if and only if

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

if and only if

G is ambiguous