RICE's Theorem

Undecidable problems:

- L is empty?
- L is regular?
- L has size 2?

This can be generalized to all non-trivial properties of Turing-acceptable languages

Non-trivial property:

A property P possessed by some Turing-acceptable languages but not all

Example: P_1 : L is empty?

YES $L = \emptyset$

NO $L = \{\text{Louisiana}\}$

NO $L = \{\text{Baton}, \text{Rouge}\}$

More examples of non-trivial properties:

P_2 : L is regular?

YES $L = \varnothing$

YES $L = \{a^n : n \geq 0\}$

NO $L = \{a^n b^n : n \geq 0\}$

P_3 : L has size 2?

NO $L = \varnothing$

NO $L = \{\text{Louisiana}\}$

YES $L = \{\text{Baton}, \text{Rouge}\}$

Trivial property:

A property P possessed by ALL Turing-acceptable languages

Examples: P_4 : L has size at least 0?

True for all languages

P_5 : L is accepted by some Turing machine?

True for all Turing-acceptable languages

We can describe a property P as the set of languages that possess the property

If language L has property P then $L \in P$

Example: P : L is empty?

YES $L_1 = \emptyset$

$P = \{L_1\}$

NO $L_2 = \{\text{Louisiana}\}$

NO $L_3 = \{\text{Baton}, \text{Rouge}\}$

Example: Suppose alphabet is $\Sigma = \{a\}$

P : L has size 1?

NO $\varnothing$

YES $\{\lambda\}$ $\{a\}$ $\{aa\}$ $\{aaa\}$ $\cdots$

NO $\{\lambda, a\}$ $\{\lambda, aa\}$ $\{a, aa\}$ $\cdots$

NO $\{\lambda, a, aa\}$ $\{aa, aaa, aaaa\}$ $\cdots$

$$P = \{\{\lambda\}, \{a\}, \{aa\}, \{aaa\}, \{aaaa\}, \ldots\}$$

Non-trivial property problem

Input: Turing Machine M

Question: Does $L(M)$ have the non-trivial property P? $\quad L(M) \in P$?

Corresponding language:

$$PROPERTY_{TM} = \{\langle M \rangle : M \text{ is a Turing machine such that } L(M) \text{ has the non-trivial property } P, \text{ that is, } L(M) \in P\}$$

Rice's Theorem: $PROPERTY_{TM}$ is undecidable

(the non-trivial property problem is unsolvable)

Proof: Reduce

A_{TM} (membership problem)

to

$PROPERTY_{TM}$ or $\overline{PROPERTY_{TM}}$

We examine two cases:

Case 1: $\emptyset \in P$

Examples: P : $L(M)$ is empty?

P : $L(M)$ is regular?

Case 2: $\emptyset \notin P$

Example: P : $L(M)$ has size 2?

Case 1: $\emptyset \in P$

Since P is non-trivial, there is a Turing-acceptable language X such that: $X \notin P$

Let M_X be the Turing machine that accepts X

Reduce

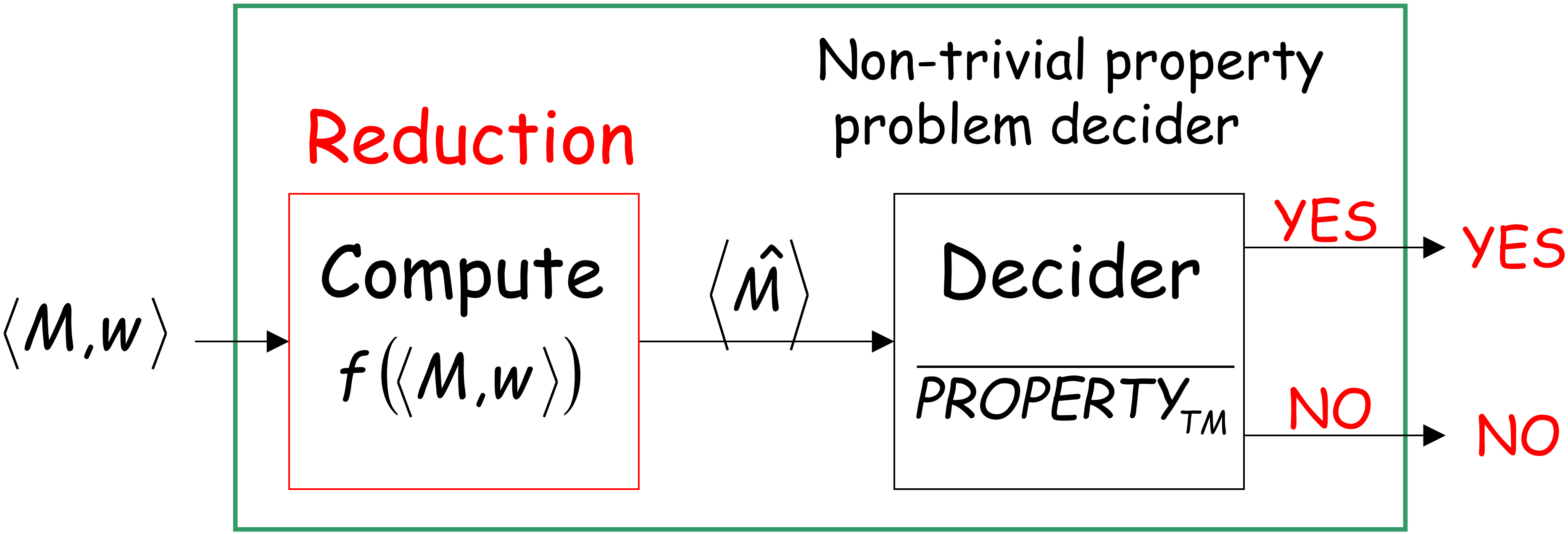
A_{TM} (membership problem)

to

$\overline{PROPERTY_{TM}}$

membership problem decider

Decider for A_{TM}



Given the reduction,
if $\overline{PROPERTY_{TM}}$ is decidable,
then A_{TM} is decidable

A contradiction!
since A_{TM}
is undecidable

We only need to build the reduction:

Reduction

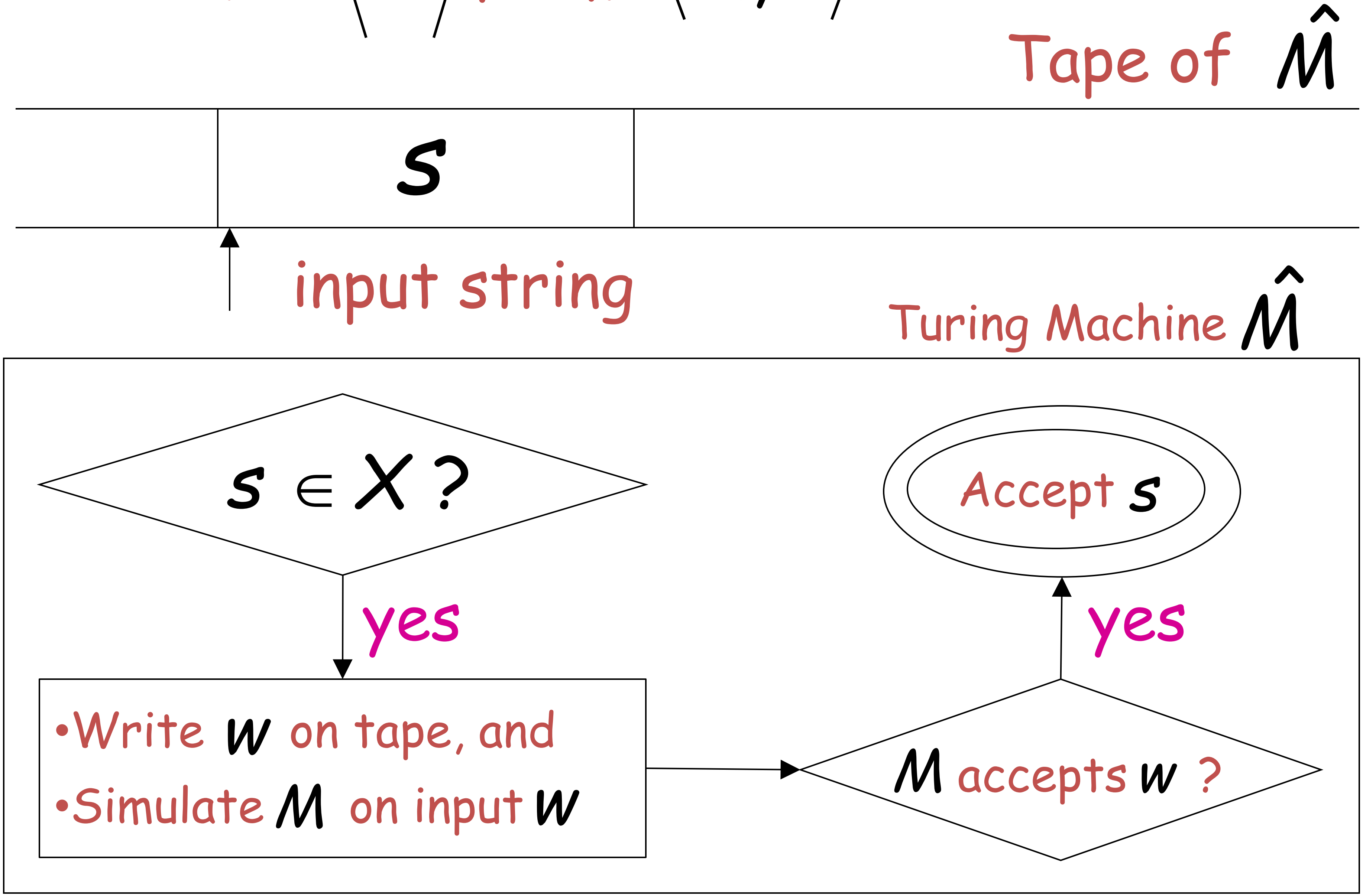
$$\langle M,w \rangle \longrightarrow \boxed{\text{Compute } f(\langle M,w \rangle)} \longrightarrow \langle \hat{M} \rangle$$

$$= f(\langle M,w \rangle)$$

So that:

$$\langle M,w \rangle \in A_{TM} \iff \langle \hat{M} \rangle \in \overline{PROPERTY_{TM}}$$

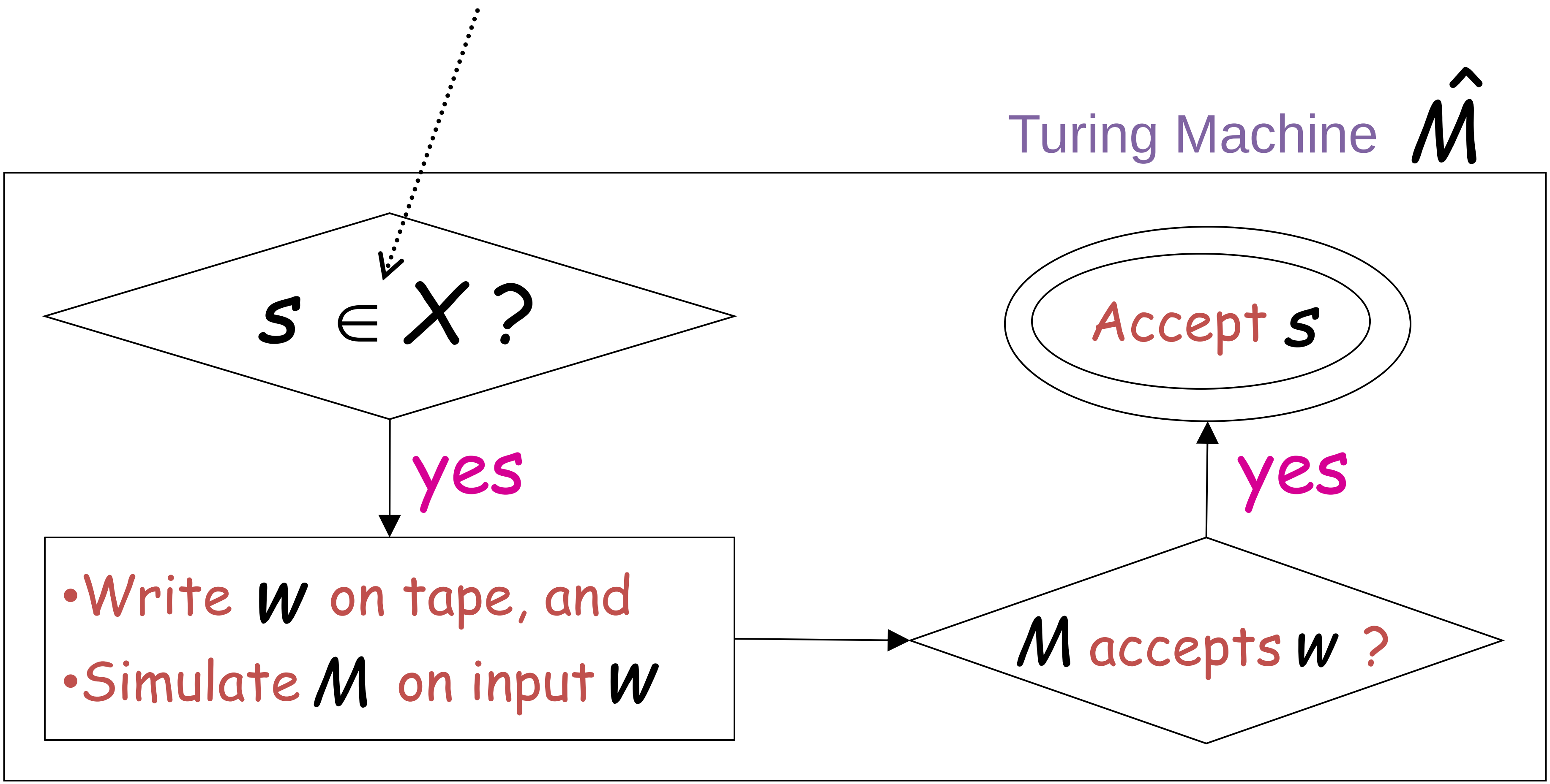
Construct $\langle \hat{M} \rangle$ from $\langle M, w \rangle$:



For this we can run machine M_X ,

that accepts language X ,

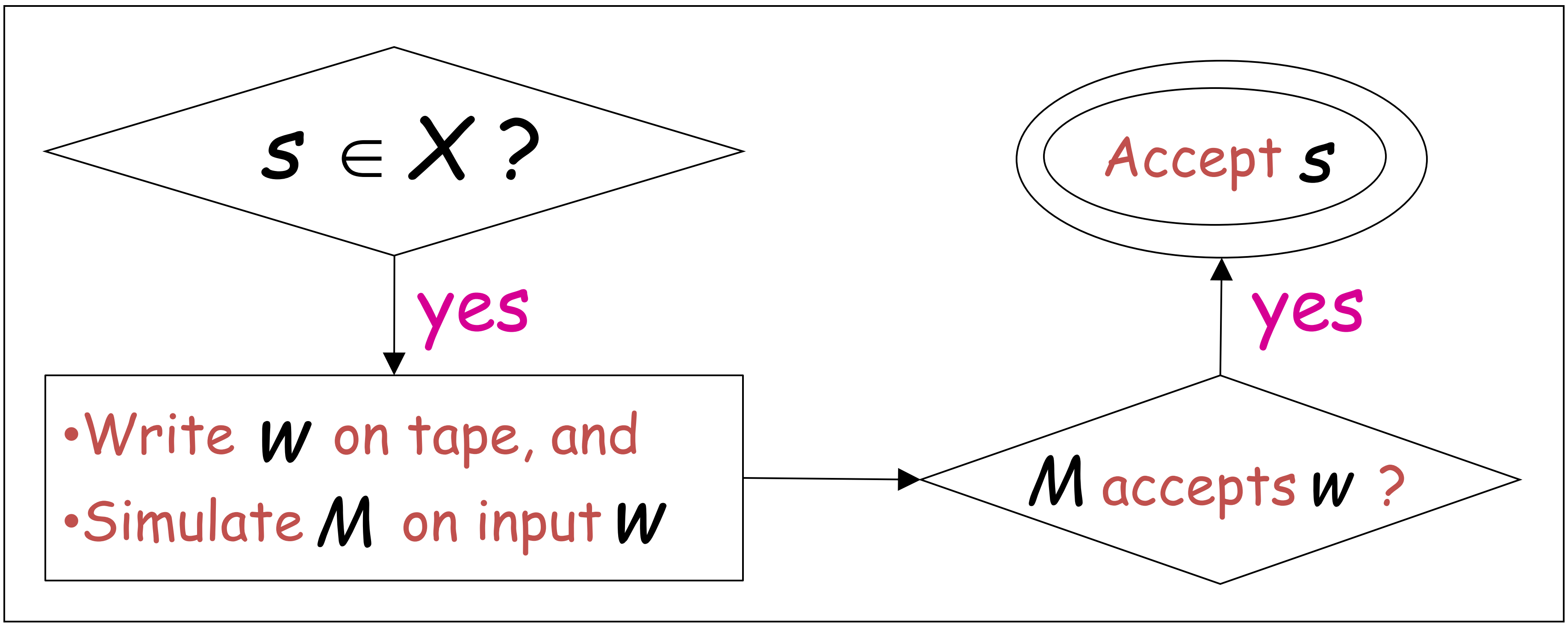
with input string s



M accepts w $\Rightarrow$ $L(\hat{M}) = X \notin P$

M does not accept w $\Rightarrow$ $L(\hat{M}) = \emptyset \in P$

Turing Machine $\hat{M}$



Therefore:

M accepts w $\Longleftrightarrow$ $L(\hat{M}) \notin P$

Equivalently:

$\langle M, w \rangle \in A_{TM}$ $\Longleftrightarrow$ $\langle \hat{M} \rangle \in \overline{PROPERTY_{TM}}$

Case 2: $\emptyset \notin P$

Since P is non-trivial, there is a Turing-acceptable language X such that: $X \in P$

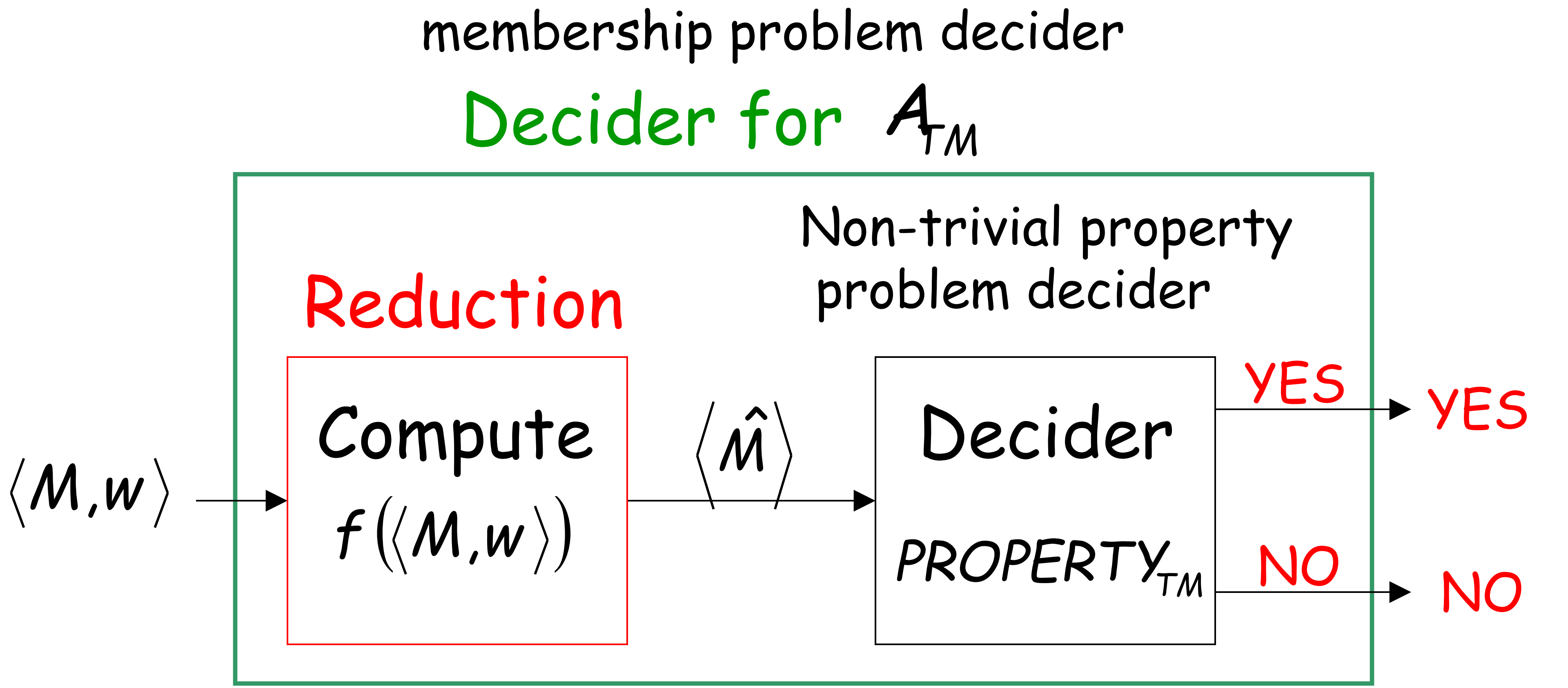
Let M_X be the Turing machine that accepts X

Reduce

A_{TM} (membership problem)

to

$PROPERTY_{TM}$



Given the reduction,
if $PROPERTY_{TM}$ is decidable,
then A_{TM} is decidable

A contradiction!
since A_{TM}
is undecidable

We only need to build the reduction:

Reduction

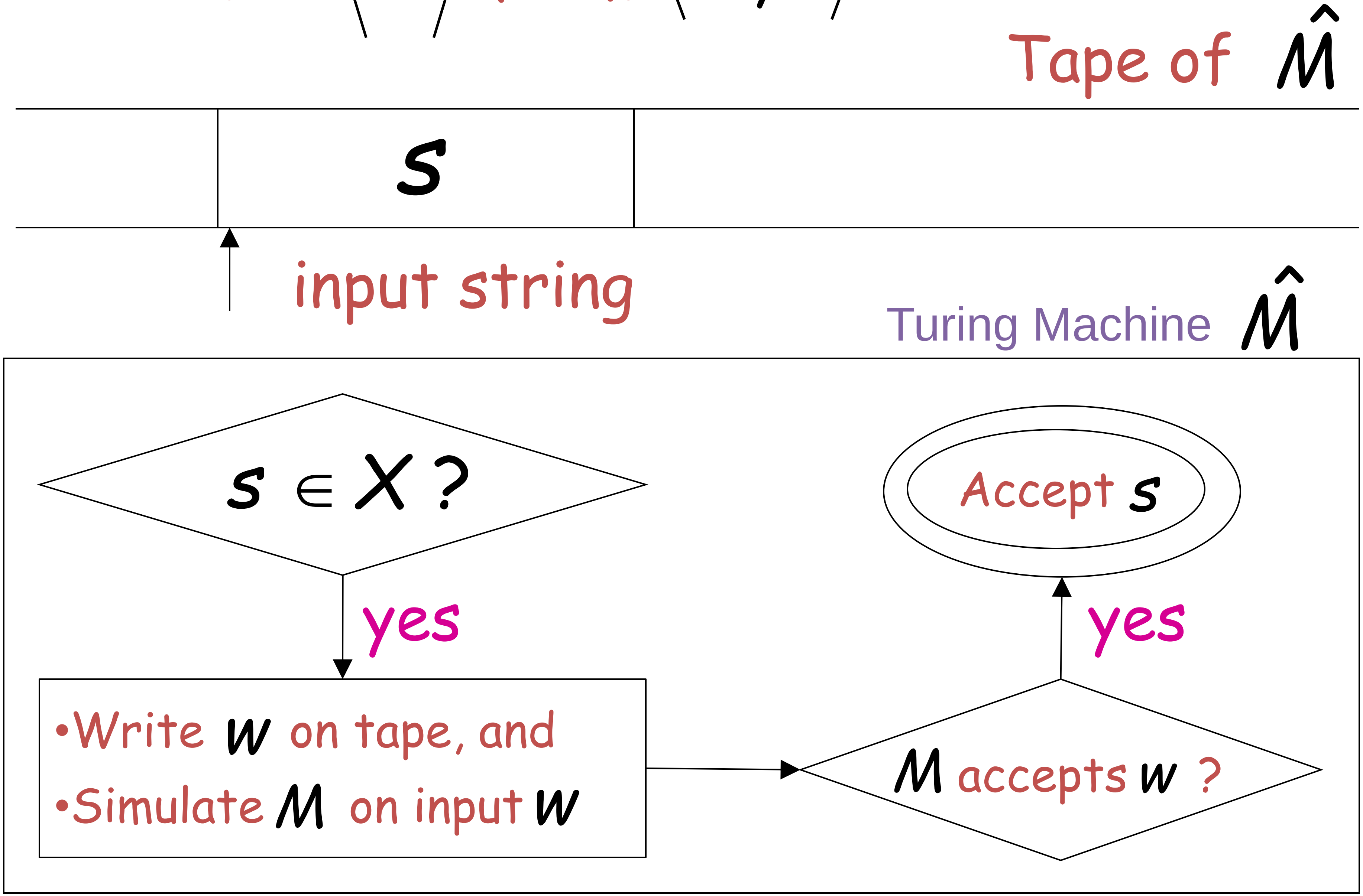
$$\langle M,w \rangle \longrightarrow \boxed{\text{Compute } f(\langle M,w \rangle)} \longrightarrow \langle \hat{M} \rangle$$

$$= f(\langle M,w \rangle)$$

So that:

$$\langle M,w \rangle \in A_{TM} \iff \langle \hat{M} \rangle \in PROPERTY_{TM}$$

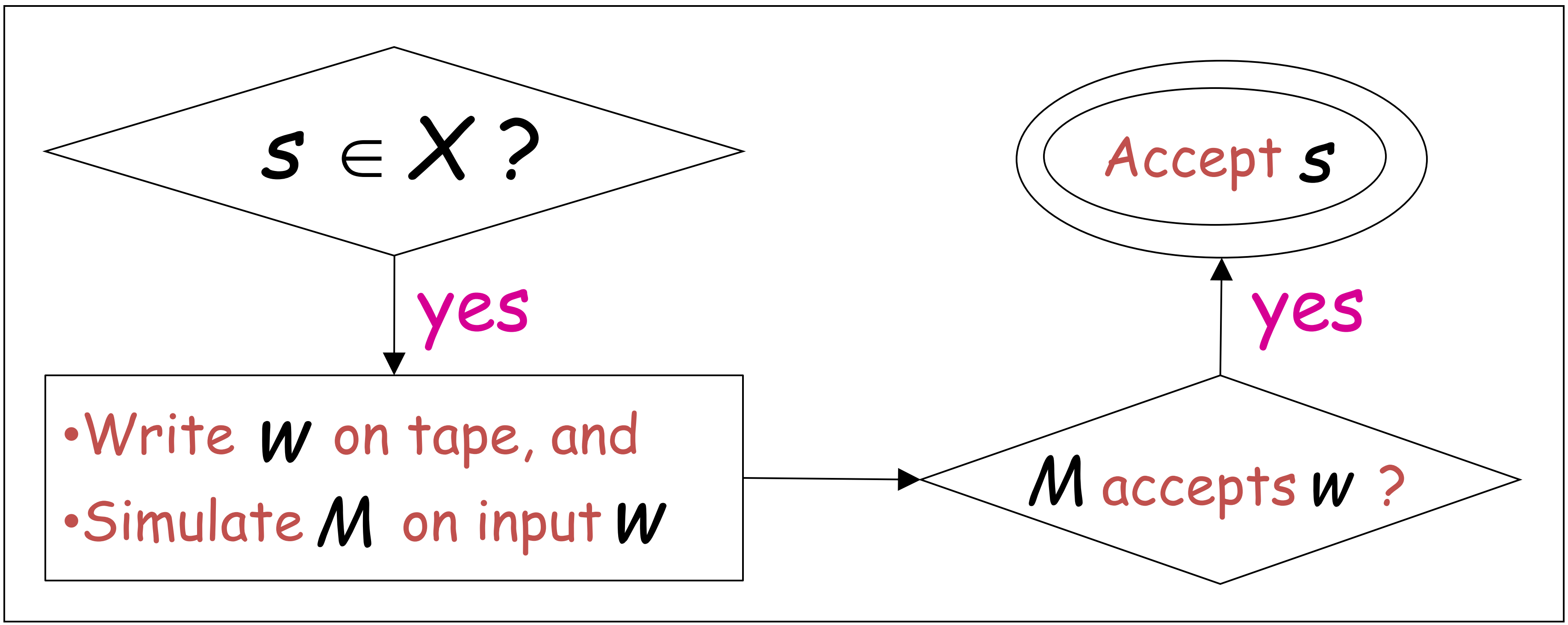
Construct $\langle \hat{M} \rangle$ from $\langle M, w \rangle$:



M accepts w $\Longrightarrow$ $L(\hat{M}) = X \quad \in P$

M does not accept w $\Longrightarrow$ $L(\hat{M}) = \varnothing \quad \notin P$

Turing Machine $\hat{M}$



Therefore:

M accepts w $\Longleftrightarrow$ $L(\hat{M}) \in P$

Equivalently:

$\langle M, w \rangle \in A_{TM}$ $\Longleftrightarrow$ $\langle \hat{M} \rangle \in PROPERTY_{TM}$

END OF PROOF

The Post Correspondence Problem

Some undecidable problems for context-free languages:

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

  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: ?

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

PC-solution: ?

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: ?

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 $\qquad w_1 w_3 w_2 = v_1 v_3 v_2$

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

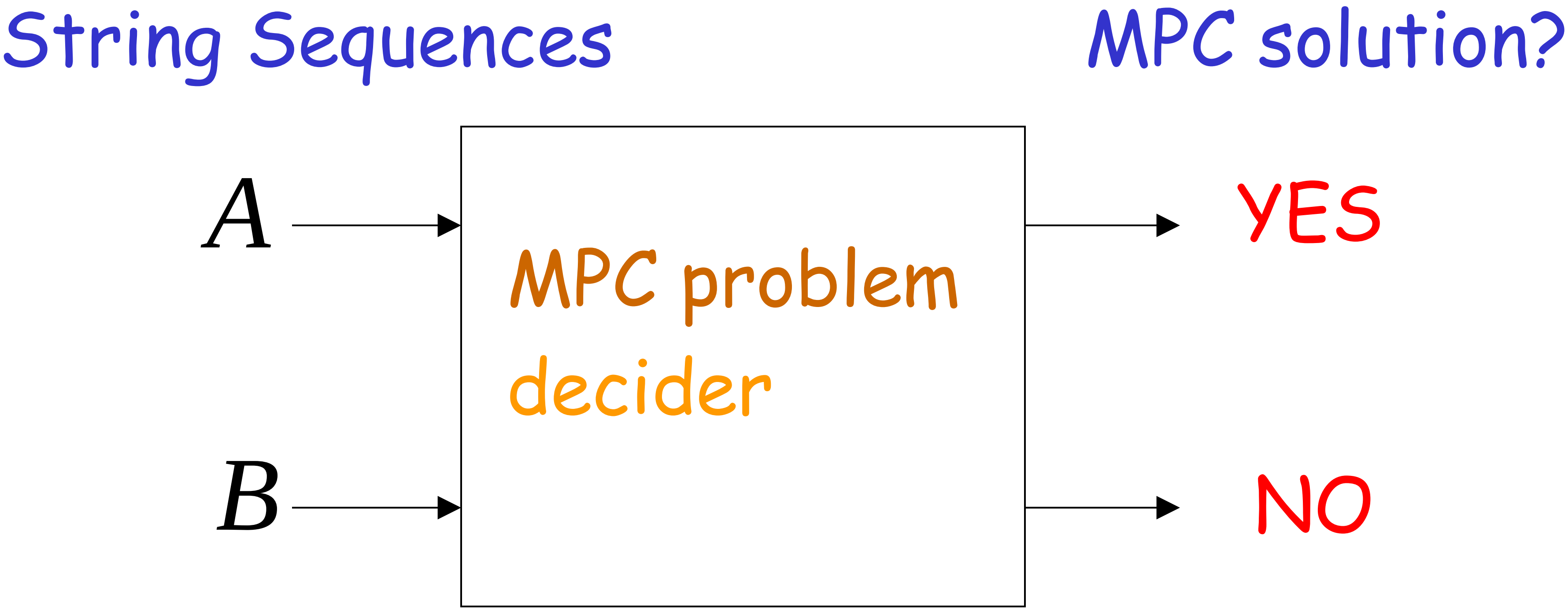
Input: unrestricted grammar G

string w

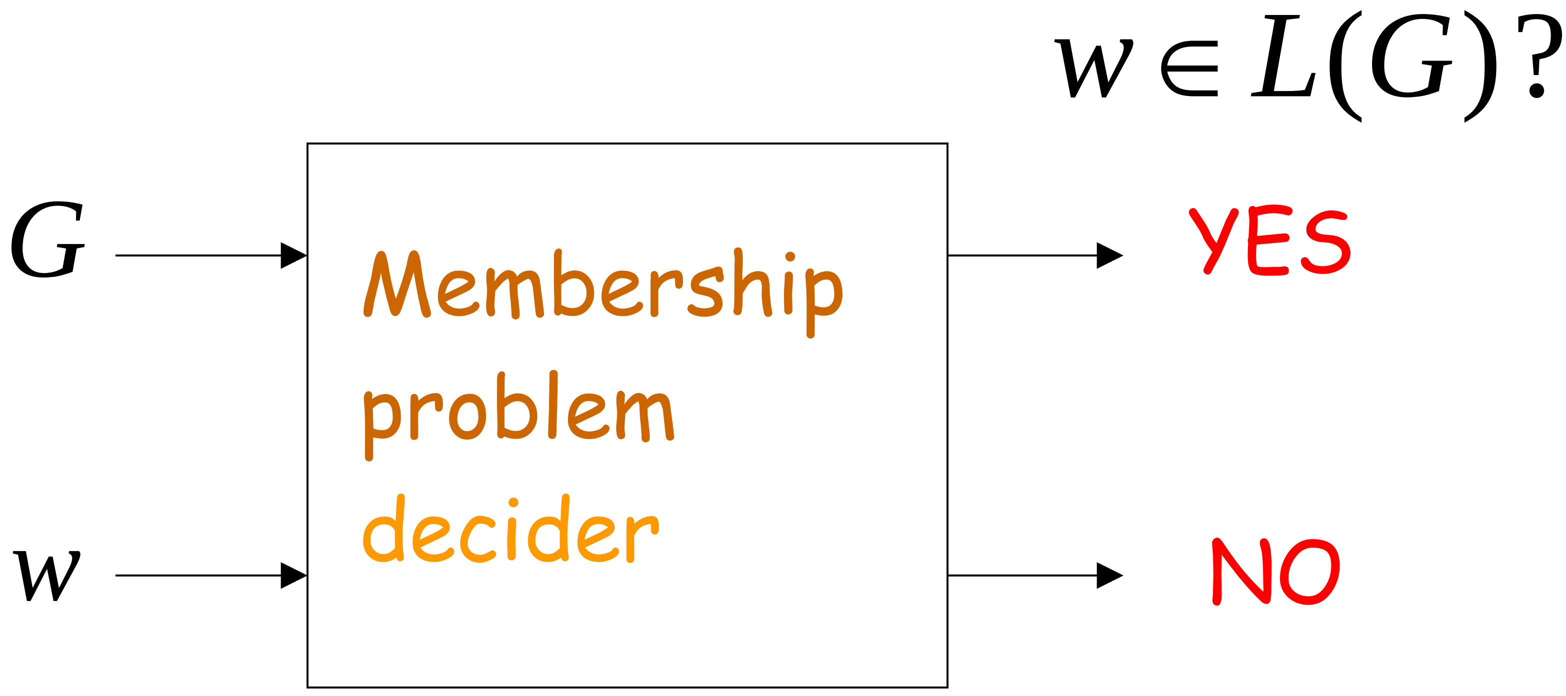
Question: $w \in L(G)$?

Undecidable

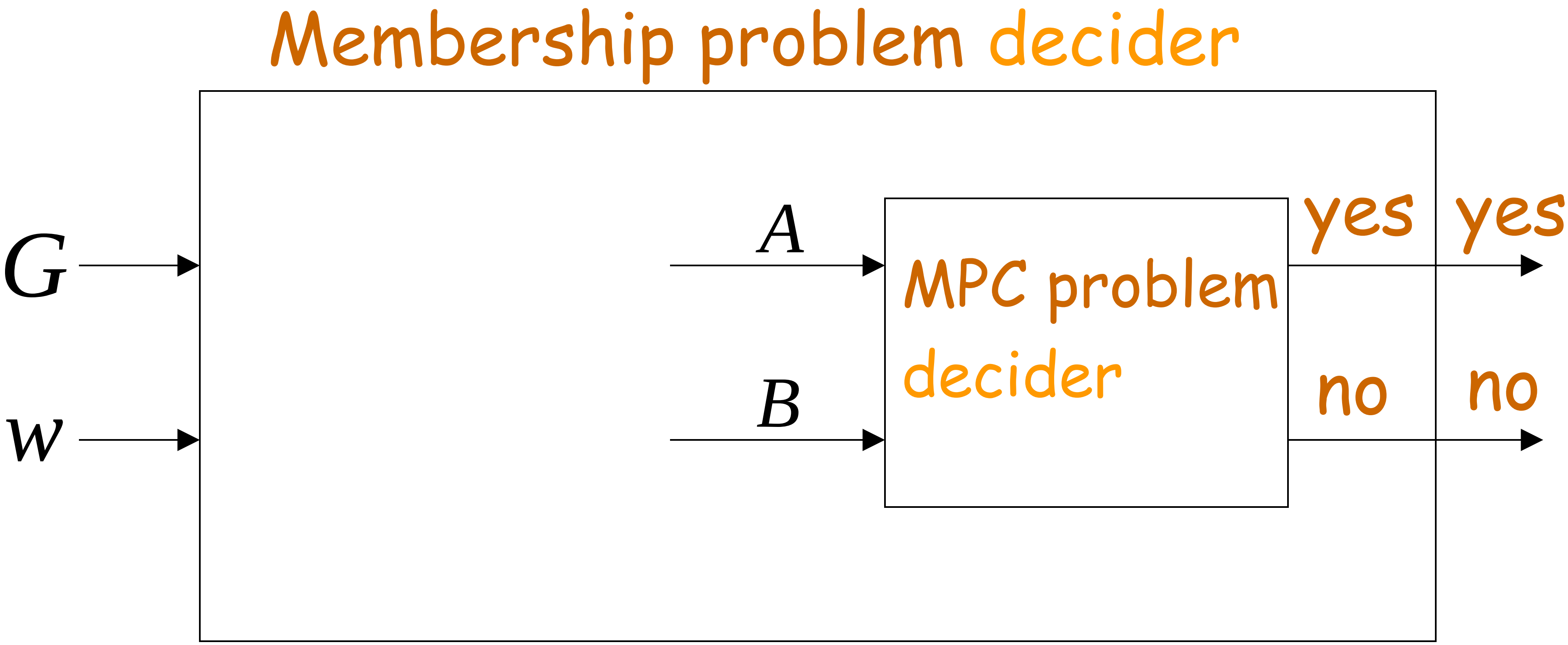
Suppose we have a decider for the MPC problem



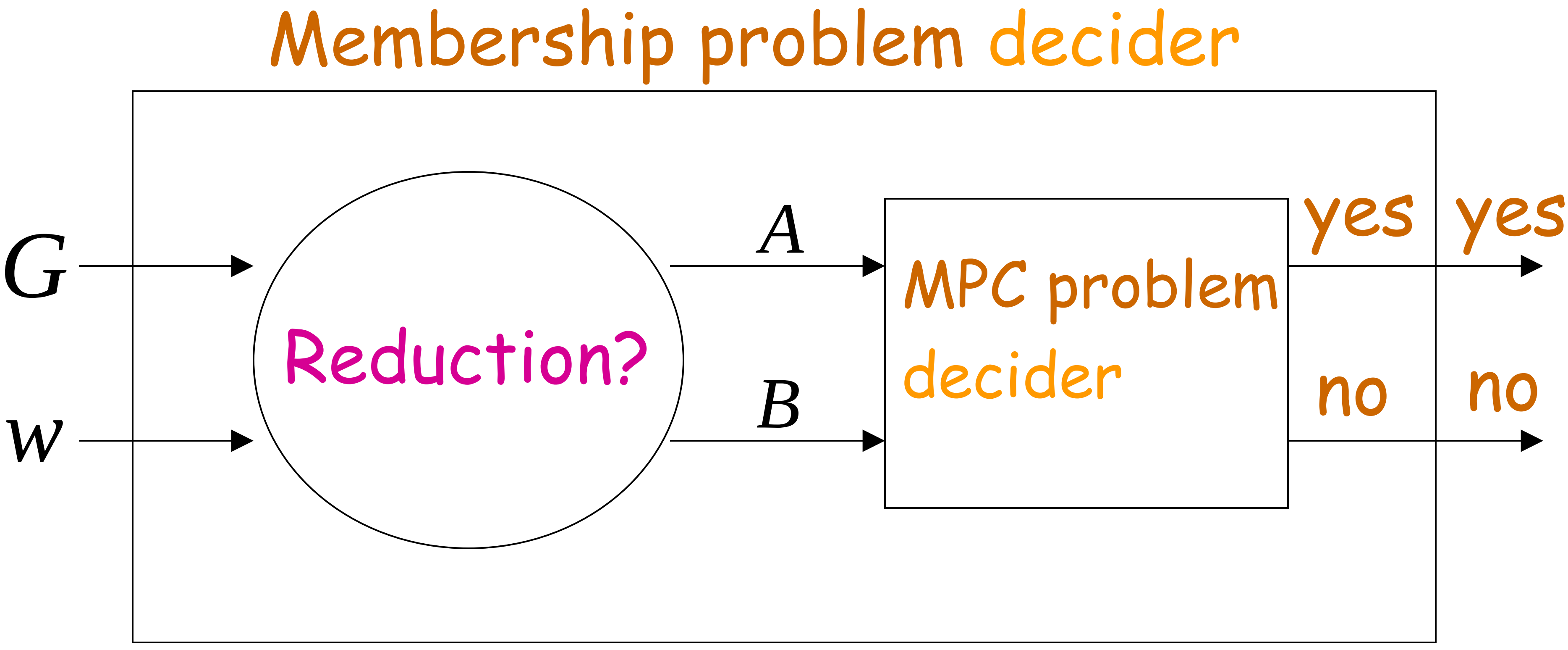
We will build a decider for the membership problem



The reduction of the membership problem to the MPC problem:



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



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
	Bbb		S
$\vdots$		$\vdots$	
	C		Bb
	aac		AC
$w_{14} :$	$\Rightarrow$	$v_{14} :$	$\Rightarrow$

Grammar G :

$$S \to aABb \mid Bbb$$

$$Bb \to C$$

$$AC \to aac$$

$aaac \in L(G)$:

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

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

Derivation: S

$$A: \quad \overbrace{F \quad S \Rightarrow}^{w_1}$$
$$B: \quad \underbrace{F}_{v_1}$$

Derivation:

$$S \Rightarrow aABb$$

$$S \to aABb \mid Bbb$$

$$Bb \to C$$

$$AC \to aac$$

A :

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

$$\underbrace{F}_{v_1}\ \underbrace{S}_{v_{10}}$$

B : $v_1\ v_{10}$

Derivation:

$$S \Rightarrow aABb \Rightarrow aAC$$

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

A :

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

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

B :

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:

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

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

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

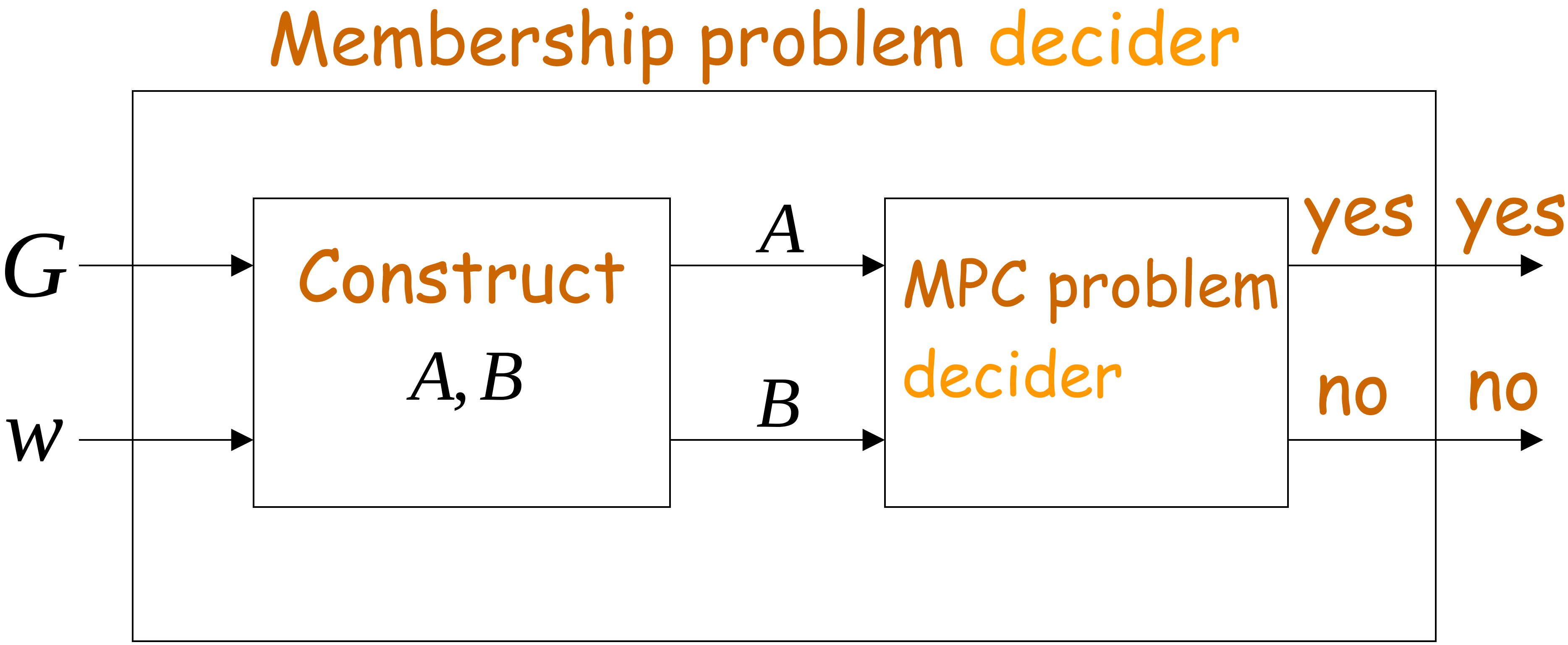
$$F \; S \Rightarrow a \; A \; B \; b \Rightarrow a \; A \; C \Rightarrow a \; a \; a \; c \; E$$

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

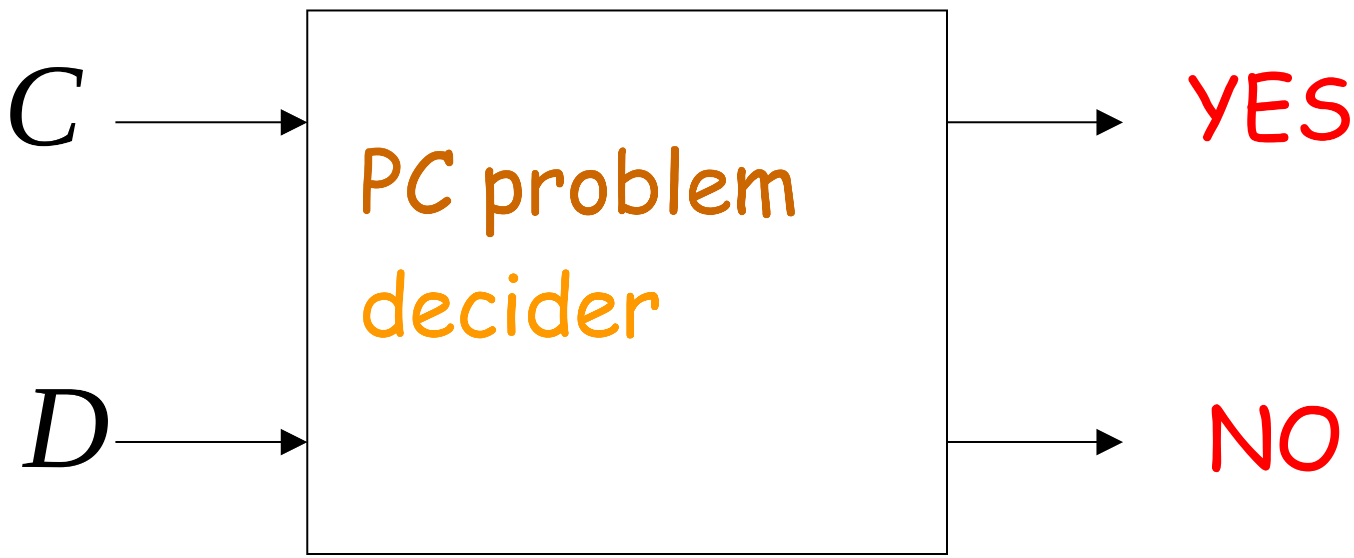
END OF PROOF

Theorem: The PC problem is undecidable

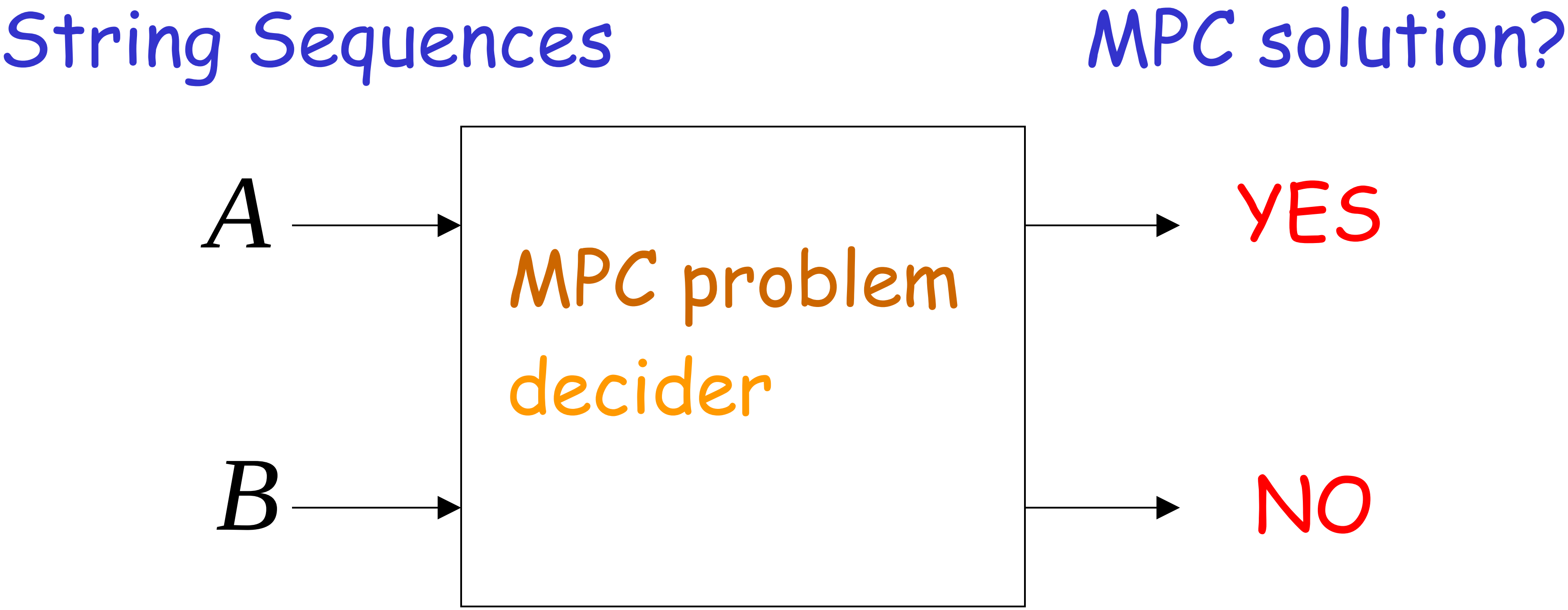
Proof: We will reduce the MPC problem to the PC problem

Suppose we have a decider for the PC problem

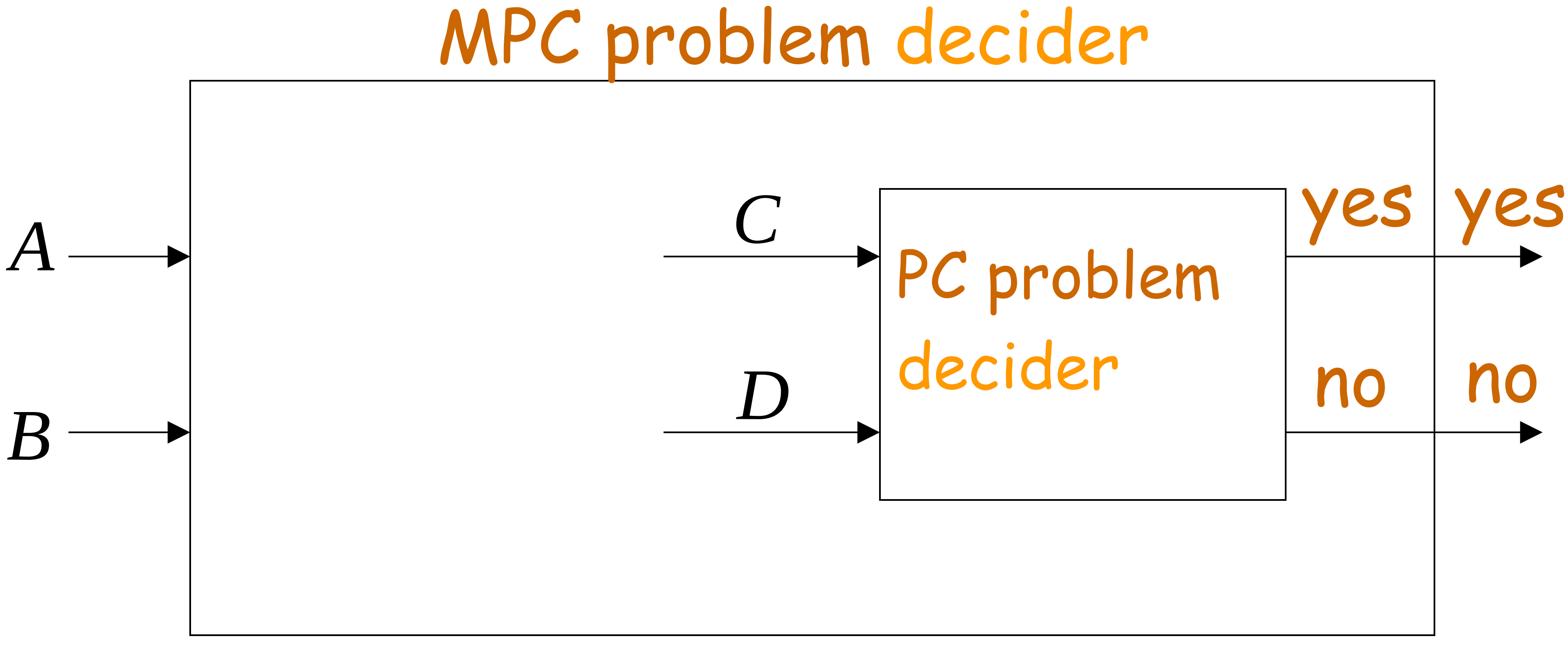
String Sequences PC solution?



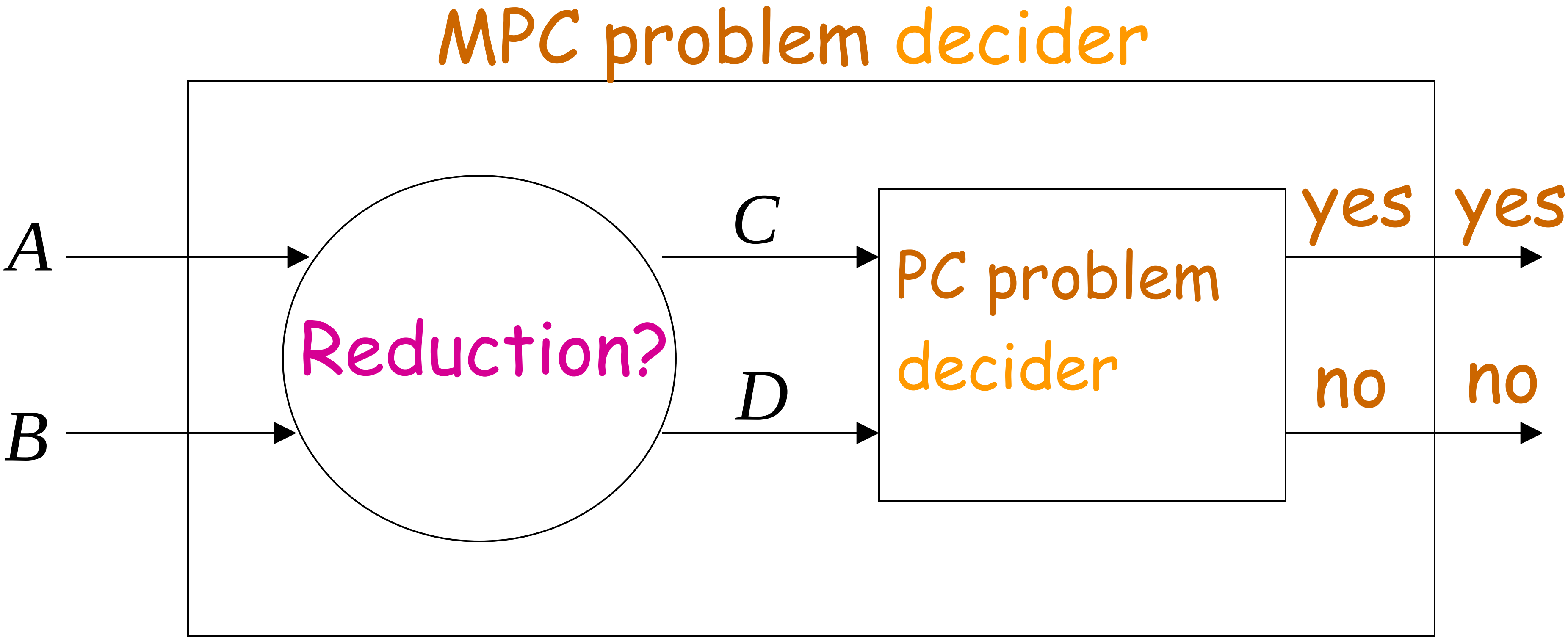
We will build a decider for the MPC problem



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

$w_i = \sigma_1 \sigma_2 \cdots \sigma_k$ $\Longrightarrow$

For each i

C

$w_i' = \sigma_1 * \sigma_2 * \cdots \sigma_k *$

replace $w_1' = * w_1'$

$w_{n+1}' = \Diamond$

B

$v_i = \pi_1 \pi_2 \cdots \pi_k$ $\Longrightarrow$

For each i

D

$v_i' = * \pi_1 * \pi_2 * \cdots * \pi_k$

$v_{n+1}' = * \Diamond$

PC-solution

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

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

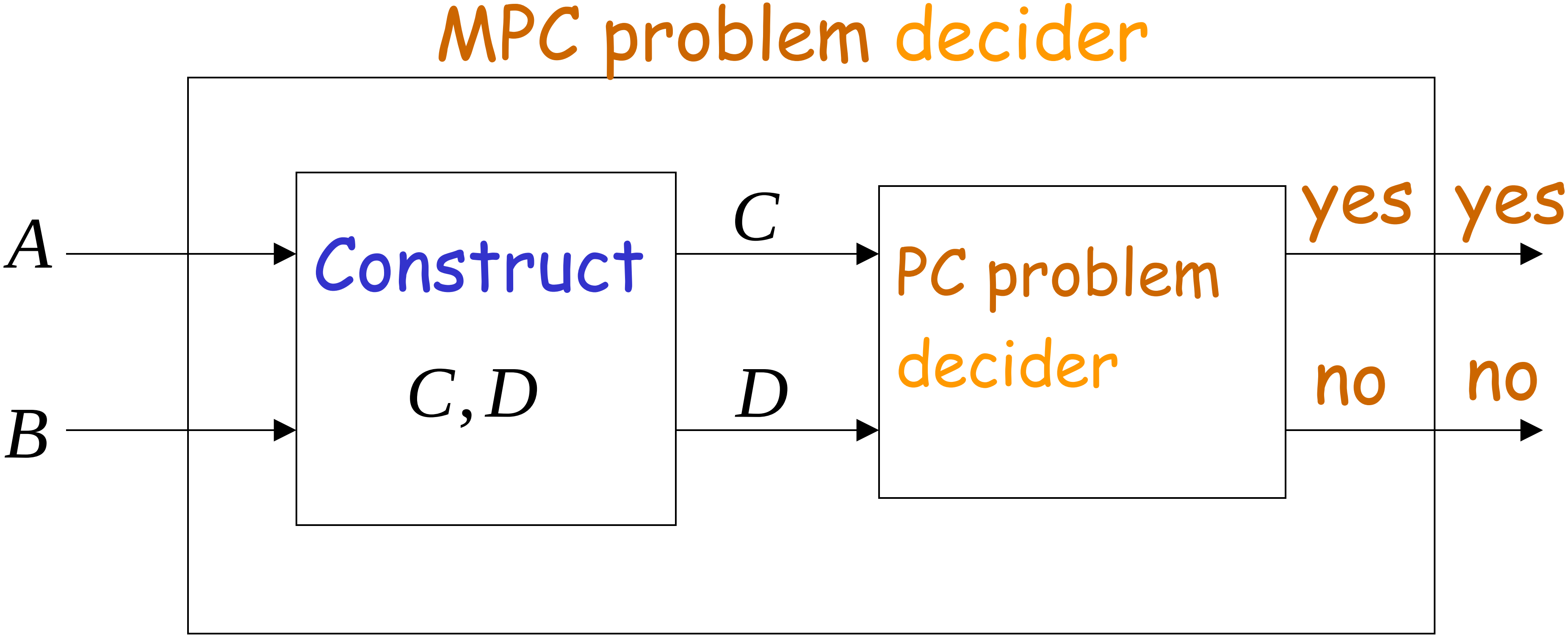
$$w_1 w_i \cdots w_k = v_1 v_i \cdots v_k$$

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
yes
no
no

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

END OF PROOF

Some <u>undecidable</u> problems for context-free languages:

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

  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

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

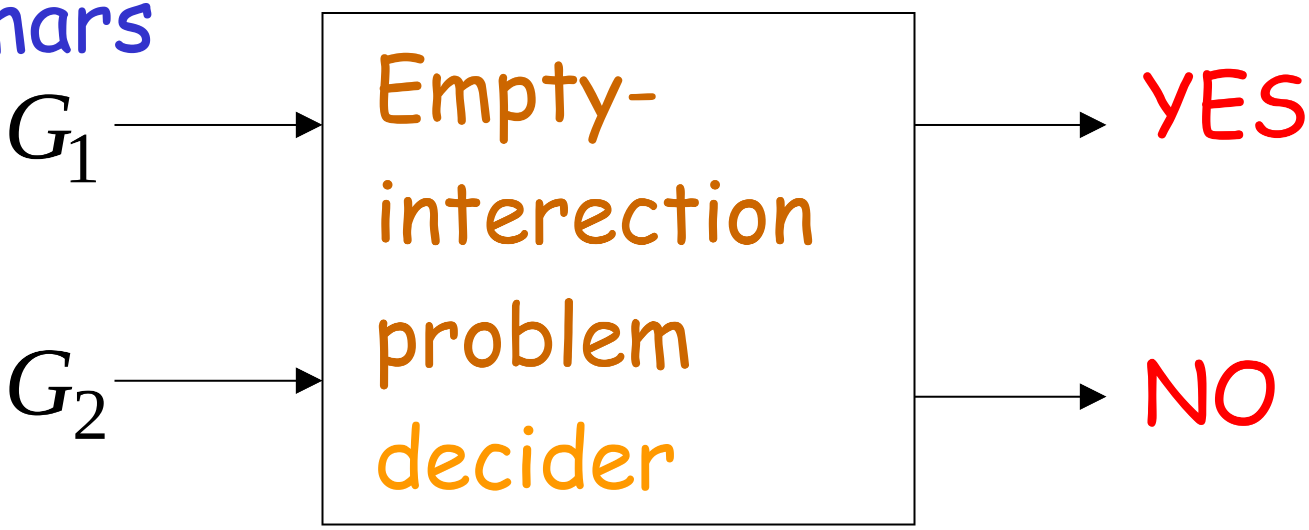
(intersection problem)

Proof: Reduce the PC problem to this problem

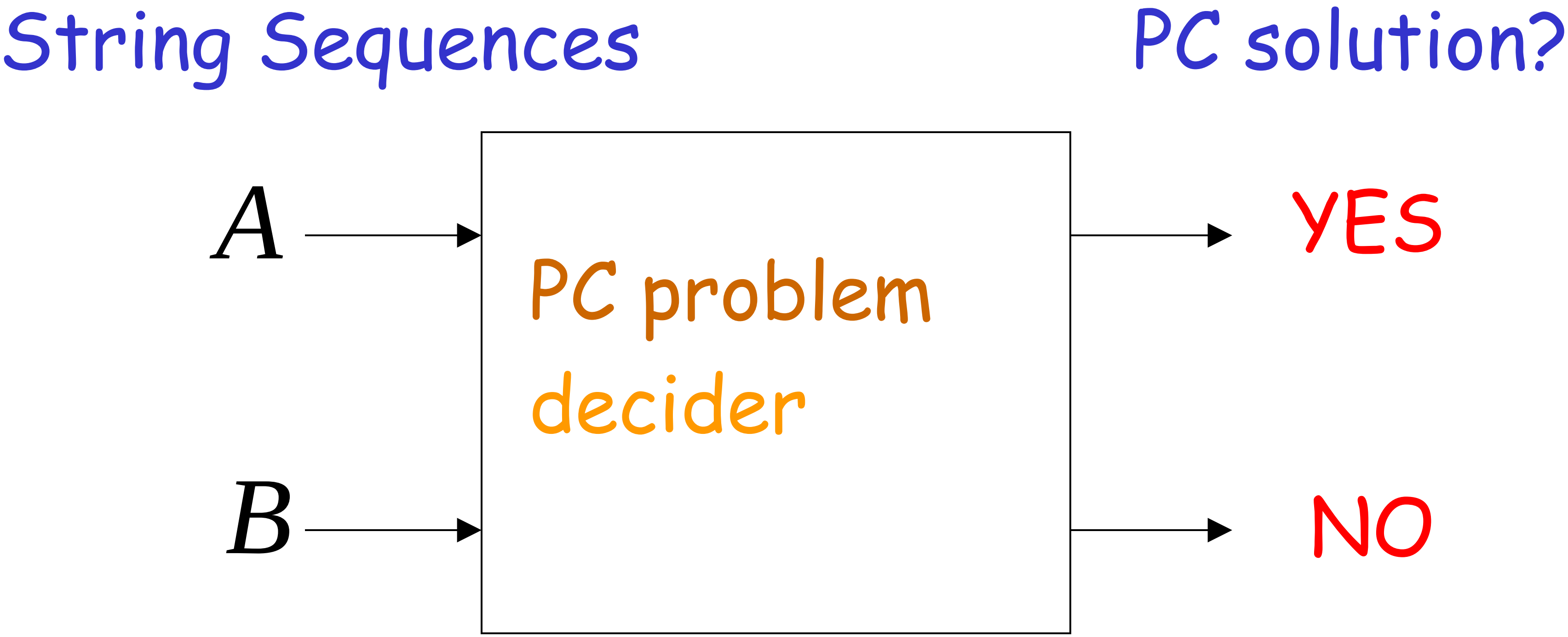
Suppose we have a decider for the intersection problem

Context-free grammars

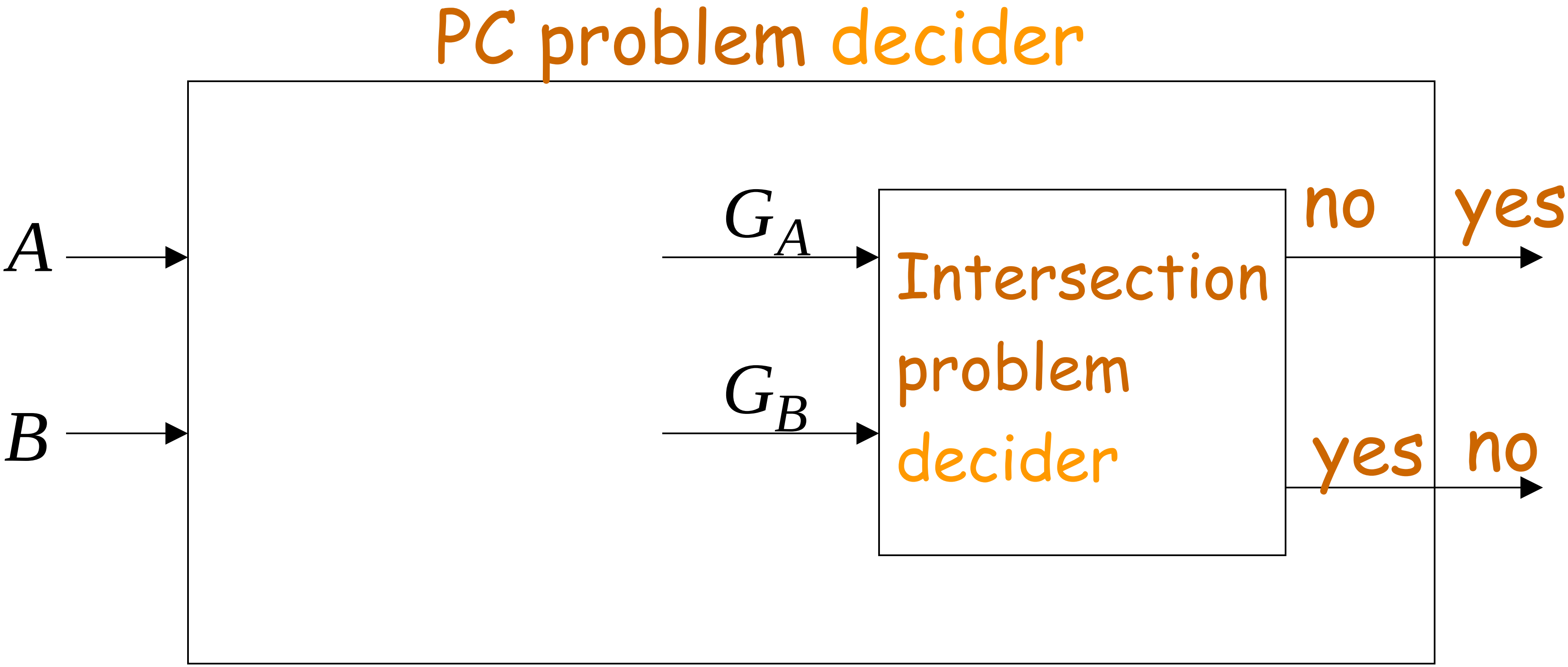
$$L(G_1) \cap L(G_2) = \emptyset ?$$



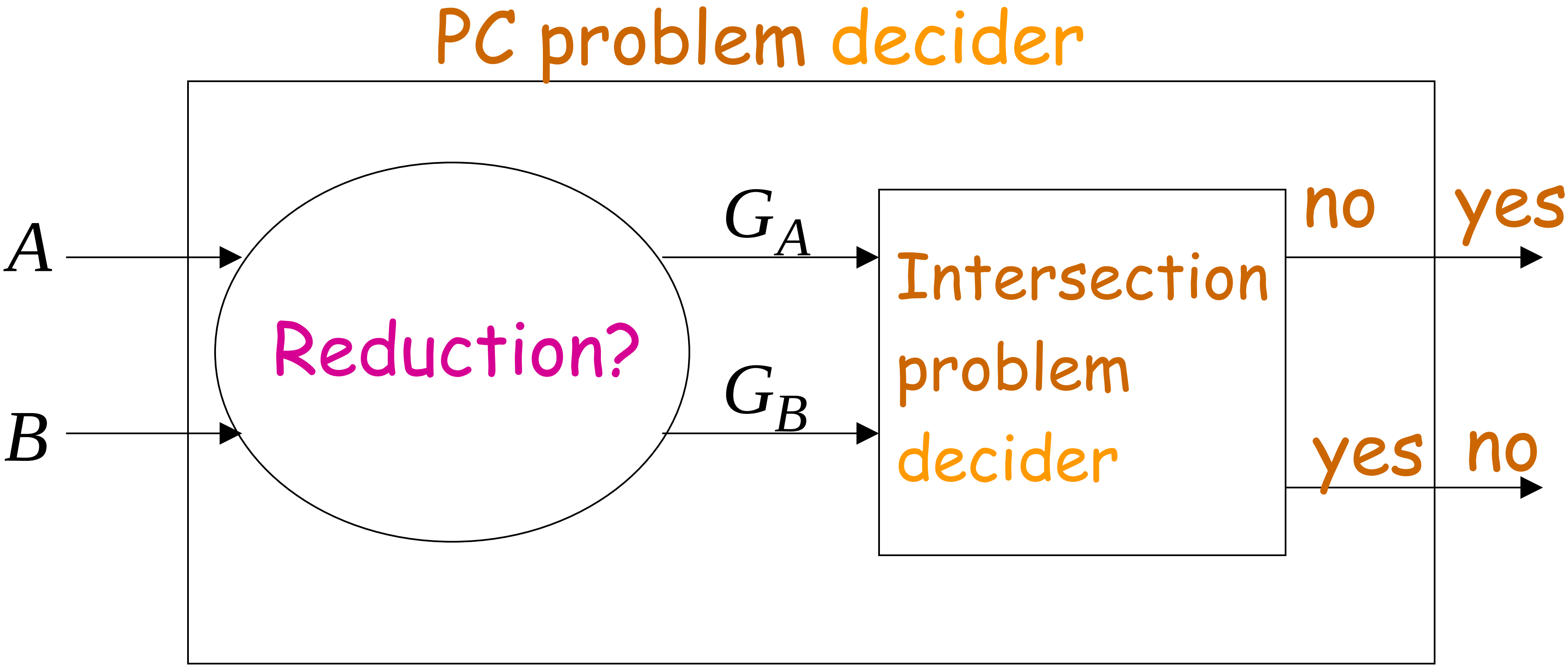
We will build a decider for the PC problem



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

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

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

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

(A, B) has a PC solution

if and
only if

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

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

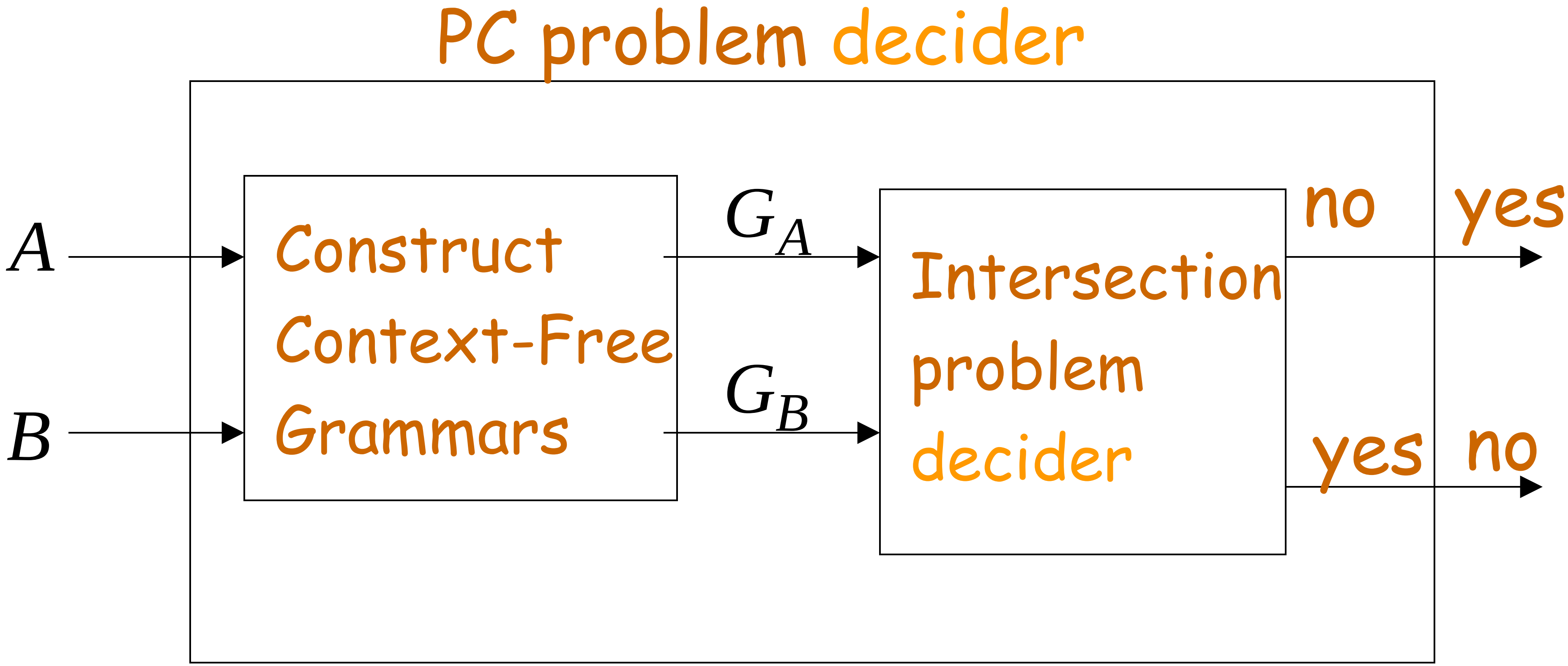
$$s = w_i w_j \cdots w_k a_k \cdots a_j a_i$$

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

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

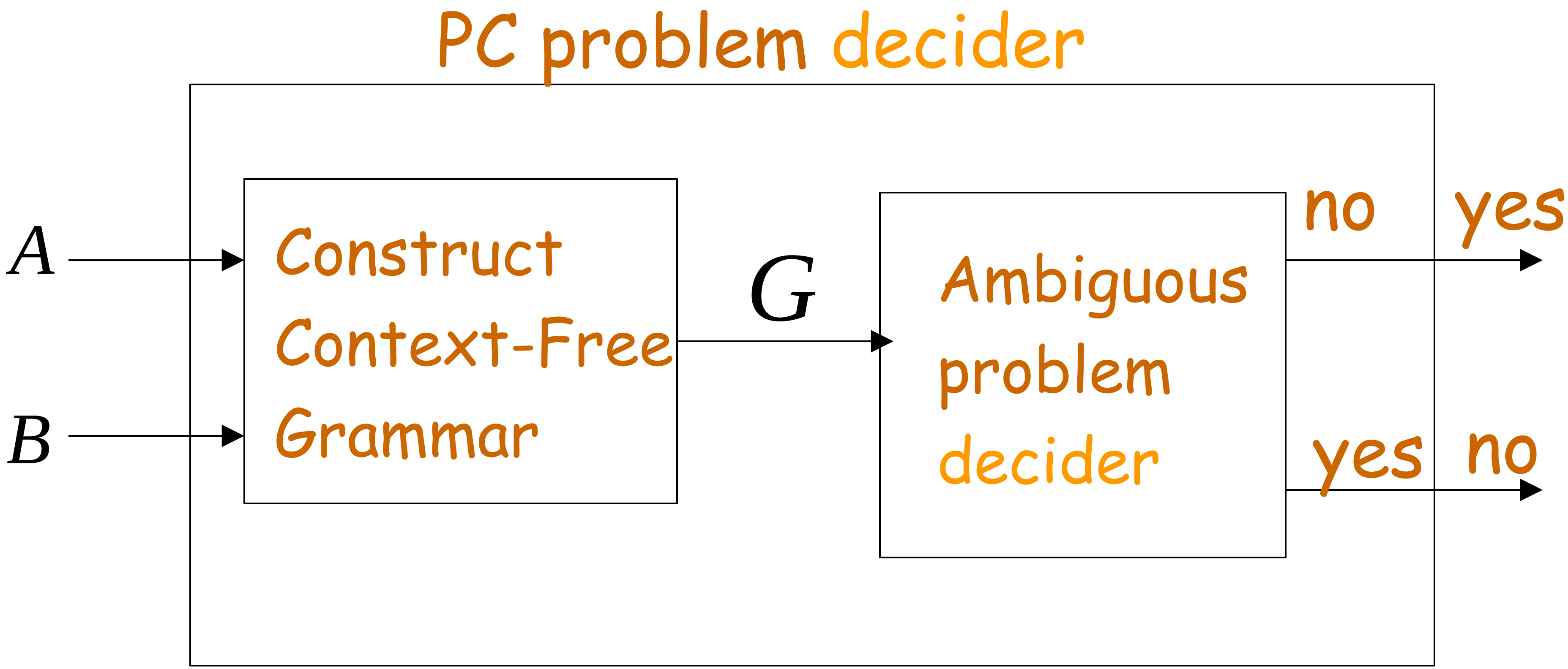
Theorem: For a context-free grammar G,

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

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

if and only if

G is ambiguous