Polynomial -time Reduction

We formalise reduction by
defining p-time reduction in terms of Turing Machines.

fast non-deterministic solutions to old yes/no problems

→ **fast non-deterministic solutions to new ones.**

Definition of p-time reduction ‘ $\leq$ ’

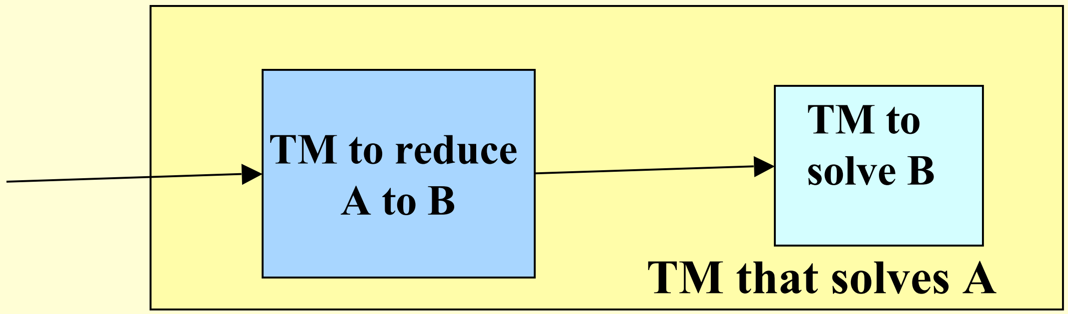
let A, B be any two yes/no problems
X a deterministic Turing Machine

X reduces A to B if: for every yes-instance w of A, $f_X(w)$ is defined and is a yes-instance of B
for every no-instance w of A, $f_X(w)$ is defined and is a no-instance of B

A reduces to B in p-time if $\exists$ a det TM X running in p-time that reduces A to B (**A $\leq$ B** if A reduces to B in polynomial time).

If A $\leq$ B and B $\leq$ A $\Rightarrow$ A ~ B.

If A ≤ B and we have a solution to B we can build a solution to A:



Example: we can reduce HCP to TSP in p-time, so HCP ≤ TSP.

This does not mean that either HCP or TSP can be solved in p-time in the general case: but it does mean that a fast (p-time) solution to TSP would facilitate a fast solution to HCP too.

TSP and HCP are in NP-complete:

each NP-complete problem reduces to all the others:

□ a p-time solution to any NP-complete problem would give a p-time solution to all the others (so putting them all into P)

≤ is a pre-order

A ≤ B we can use a fast solution to B to solve A fast..

≤ can be interpreted as "is no harder than"

≤ is a pre-order: a reflexive, transitive, binary relation.

To prove that ≤ is a pre=order on the class of yes/no problems:
Show:

1. ≤ is reflexive: ..show that for any yes/no problem A, A ≤ A
   …that there is a deterministic p-time TM which reduces A to A.
2. ≤ is transitive:..show that if A ≤ B and B ≤ C then A ≤ C.
   …that there is a deterministic p-time TM which reduces A to C

1. ≤ is reflexive: A ≤ A

let I : alphabet used to write all instances of A

let **X = ({q0}, I $\cup$ {^}, I, q0, ∅, {q0})**

be an ordinary deterministic TM

X Halts and Succeeds with no actions

output = input (unchanged)

so if w is a yes-instance of A

$\Rightarrow$ $f_X(w)$ is a yes-instance of A

if w is a no-instance of A

$\Rightarrow$ $f_X(w)$ is a no-instance of A.

$\therefore$ X reduces A to A.

$time_X(n) = 0$. **So ≤ is reflexive**

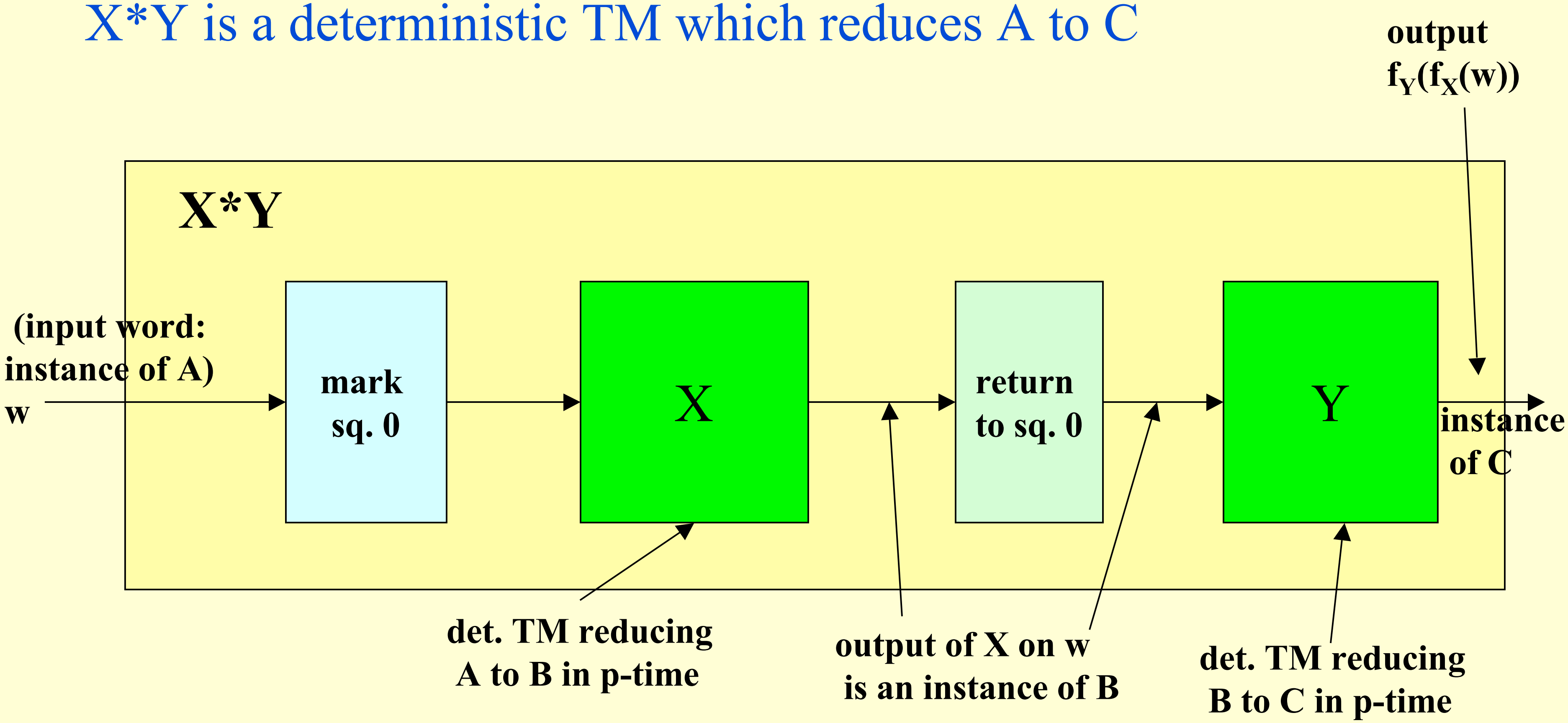
2. ≤ is transitive

suppose A ≤B and B ≤ C,

A, B, C all yes-no problems

Prove A ≤ C.

X*Y is a deterministic TM which reduces A to C



Running time of X*Y reducing A to C:

X reduces A to B in p-time…suppose $time_X(n) \leq p(n)$

Y reduces B to C in p-time….suppose $time_Y(n) \leq q(n)$
where p and q are polynomials in n

The running time of X*Y on input w of length n:

1. Mark sq. 0 — 1
2. Run X on w — $\leq p(n)$
3. Return to sq. 0 and unmark it — $\leq p(n)$
4. Run Y on $f_X(w)$ — $\leq q(p(n))$

Total run time $\leq 1 + p(n) + p(n) + q(p(n))$
..a polynomial in n.

So X*Y reduces A to C in p-time → **≤ is transitive**

NP is closed under p-time reduction

if A ≤ B we can use a fast solution to B to solve A fast.
If B is solved a by a p-time NDTM, that is B ∈ NP

..then is A ∈ NP?

Prove: if A ≤ B and B ∈ NP then A ∈ NP.

A ≤ B ⇒ ∃ a det. TM, X which reduces A to B.
B ∈ NP ⇒ ∃ a NDTM, N which solves B.

We construct a new TM, X*N which solves A.

we need to show that:

X*N is non-deterministic and
X*N runs in p-time → A ∈ NP.

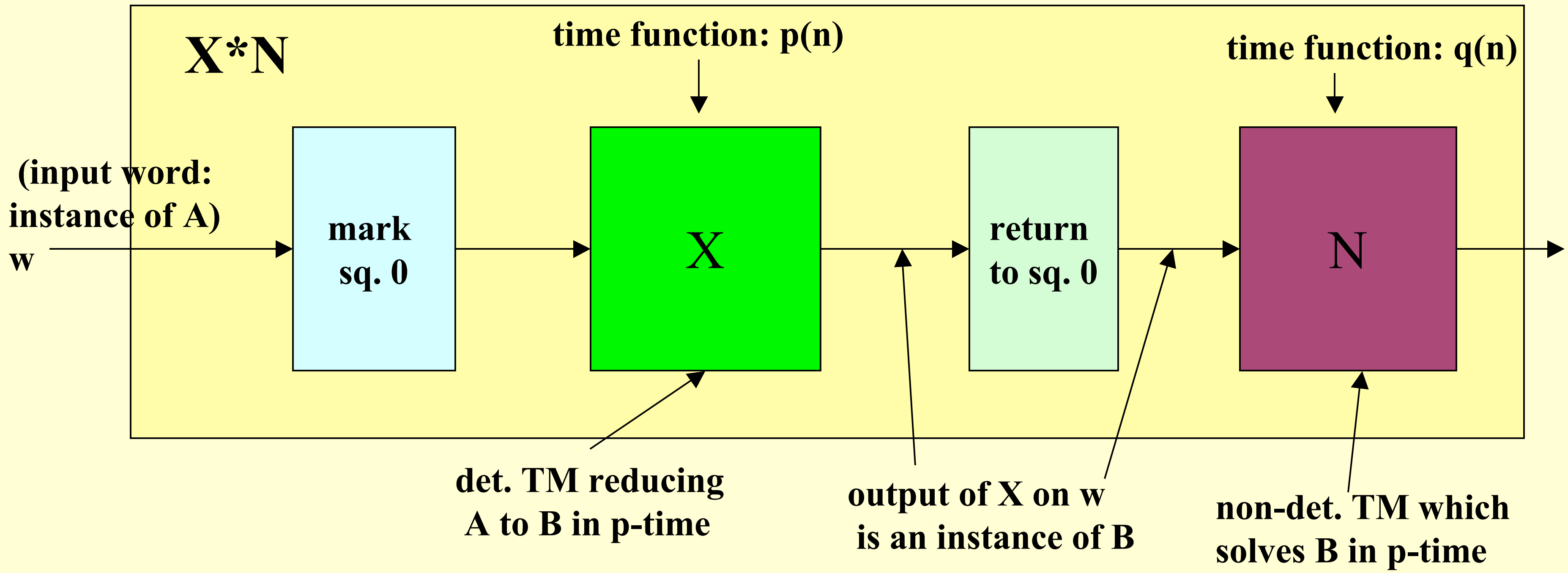
suppose $A \leq B$ and $B \in NP$, prove $A \in NP$.

X*N is a non-deterministic TM which solves A.

running time of X*N $\leq 1+p(n) + p(n)+q(p(n))$..a polynomial

$\Rightarrow$ A ∈ NP

X*N accepts a yes-instance of A
X*N rejects a no-instance of A



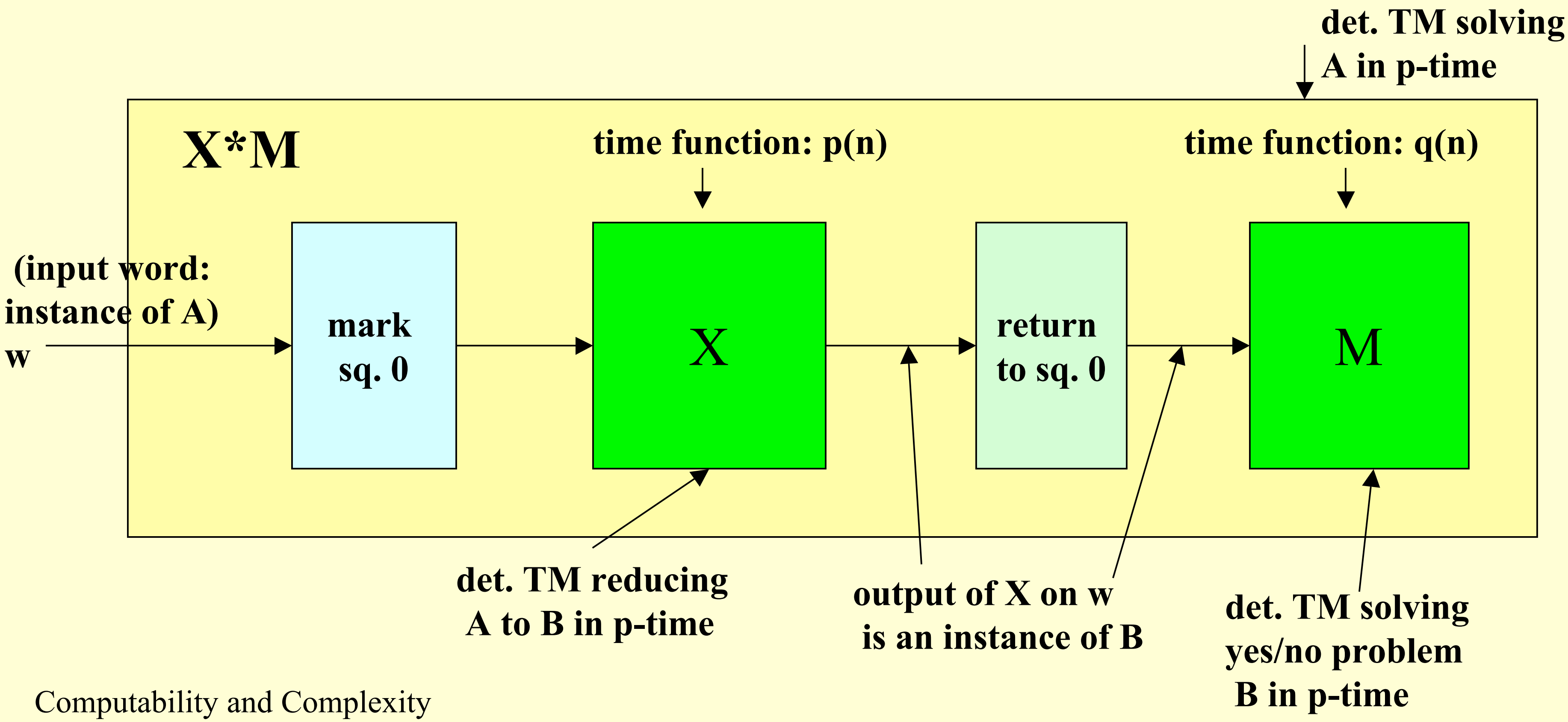
P is closed downwards under reduction

if $A \leq B$ and $B \in P$..then $A \in P$.

time function of X*M: $\text{time}_{X^*M}(n) \leq 1 + p(n) + p(n) + q(p(n))$

a polynomial

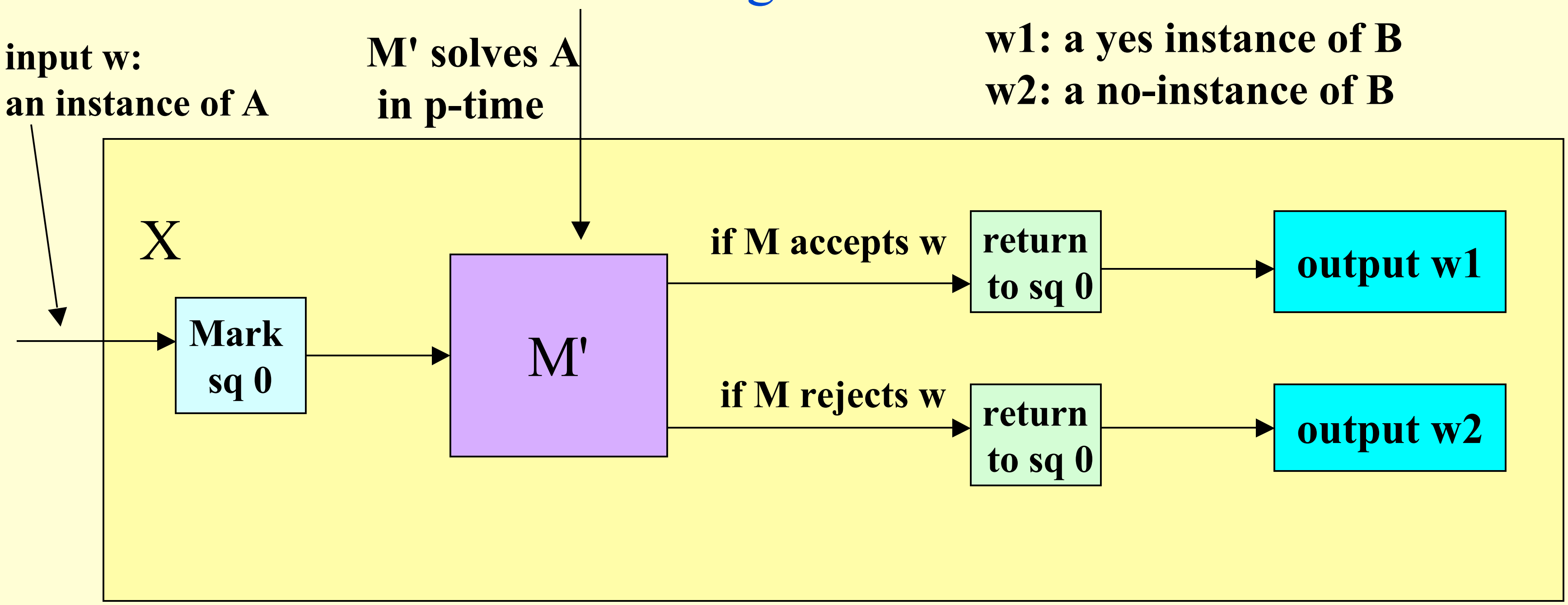
X*M is a deterministic p-time TM which solves A



The P-problems are the ≤-easiest.

(The problems in P are the easiest with respect to our difficulty ordering)

Prove that if A is any problem in P and B is any yes/no problem A ≤ B let M be a det. p-time TM which solves A: modify H&S and H&F to give M'



Summary

We have defined:

polynomial-time reduction: if A, B are yes/no problems:

A reduces to B in p-time if $\exists$ a det TM X running in p-time that reduces A to B (**A ≤ B** if A reduces to B in polynomial time).

Properties of ≤: **≤ is a pre-order**..

..a reflexive, transitive, binary relation

..permitting an ordering in difficulty or complexity of problems.

Properties of P and NP with respect to reduction:

P is closed downwards under ≤ : for yes/no problems A, B

if A ≤B and B $\in$ P..then A $\in$ P

NP is closed downwards under ≤ :for yes/no problems A, B

if A ≤B and B $\in$ NP..then A $\in$ NP

The class P is the ≤-easiest class of yes/no problems.